Two Improved Lossless Image Compression Methods

by

Mohammad Sharif Chowdhury

A Thesis Presented to the

FACULTY OF THE COLLEGE OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

ELECTRICAL ENGINEERING

December, 1998

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI

A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor MI 48106-1346 USA
313/761-4700 800/521-0600

TWO IMPROVED LOSSLESS IMAGE COMPRESSION METHODS

BY
MOHAMMAD SHARIF CHOWDHURY

A Thesis Presented to the
FACULTY OF THE COLLEGE OF GRADUATE STUDIES
KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE
In
ELECTRICAL ENGINEERING

DECEMBER 1998

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN, SAUDI ARABIA

This thesis, written by

Mohammad Sharif Chowdhury

under the direction of his thesis advisor, and approved by his thesis committee, has been presented to and accepted by the Dean, College of Graduate Studies, in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

Thesis Committee :

Chairman (Dr. Samir H. Abdul-Jauwad)

Member (Dr. Maamar Bettayeb)

Member (Dr. Saud Al-Semari)

Member (Dr. Maan Kousa)

Dr. Samir Al-Baiyat
(Department Chairman)

Dr. Abdallah M. Al-Shehri
(Dean, College of Graduate Studies)

Date : 9/2/99

كلية الدراسات العليا
COLLEGE OF GRADUATE STUDIES

Dedicated to the memory

Of

My Parents

ACKNOLEDGEMENT

All glories and praises to Almighty Allah without whose help no work can be accomplished. Acknowledgment is due to King Fahd University of Petroleum and Minerals for providing support to this research.

I am grateful to my thesis advisor Dr. Samir H. Abdul-Jauwad for his continuous guidance and assistance during the period of this research. It was his insight and knowledge of the subject matter that made this task easier. I again thanks to Dr. Samir H. Abdul-Jauwad for providing x-ray images from his own knee.

I would like to express my thanks to the thesis committee members Dr. Maamar Bettayeb, Dr. Saud Al-Semari, and Dr. Maan Kousa for their opinion, appreciation and helpful suggestions.

I wish to thank Dr. Adnan Mohammad Alattar for his advice and help while he was in KFUPM. My heartfelt thanks and gratefulness to my father who sent me here in KPUPM and left forever before the completion of my MS degree.

Special thanks to my colleagues and friends for their encouragement and various help that they provided throughout my graduate studies at KFUPM. I would like to thank my friends Mr. Hasibul Majid, Mr. Nuruzzaman, Mr. Mozahar hossain and Mr. Anowar for their help during this research.

Finally the cooperation of the department chairman Dr. Samir Al Baiyat during my critical moments is gratefully appreciated.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure ***Page***

THESIS ABSTRACT

Name	: Mohammad Sharif Chowdhury
Title	: Two Improved Lossless Image Compression Methods
Major field	: Electrical Engineering
Date of degree	: December 1998

Two different methods of lossless Image compression has been investigated in this thesis. The first method, Method-I reduces bits per pixel by modeling the prediction errors after their generation when compared to lossless JPEG. In Method-I prediction errors are mapped to bit streams of many layers and each layer represents certain prediction error. The ratio of zero to one in the different layers of the bit stream is different. In this method, bit streams are divided into different groups and each group is entropy coded instead of entropy coding the prediction errors to give the desired compression.

Method-II reduces the number of codes in the code table by grouping the prediction errors. In this method, eight codes for eight bit precision image are made based on the most significant bit position of the absolute values of the prediction error and sending the remaining bits of the absolute values of prediction error as is. Due to short code table, Method-II decodes the image much faster than JPEG and reduces the difference between the encoding and decoding time to make the system symmetric for real time application while sacrificing some compression ratio.

MASTER OF SCIENCE
KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN, SAUDI ARABIA
December 1998

خلاصة الرسالة

اسم الطالب : محمد شريف شودري

عنوان الرسالة : طريقتين محسنتين لضغط الصور دون خسارة

التخصص : هندسة كهربائية

تاريخ التخرج : شعبان ١٤١٩ ، ديسمبر ١٩٩٨

تمت دراسة طريقتين مختلفتين لضغط الصور دون خسارة في هذا البحث .

الطريقة الأولى تعتمد على تقليل عدد البيت (bits) عبر كل بيكسل (Pixel) وذلك بواسطة دراسة أخطاء التنبؤ بعد توليده ومقارنته بطريقة (JPEG) التي تتميز بعدم حدوث خسارة . وفي هذه الطريقة يتم تحويل أخطاء التنبؤ إلى سيل من البيت (bits) ذو طبقات عديدة ، وكل طبقة تمثل أخطاء التنبؤ . وتختلف نسبة الصفر إلى الواحد بين طبقة وأخرى من طبقات سيل البيت (bit stream) . وفي هذه الطريقة ، تقسم سيل البيت إلى مجموعات مختلفة وتقاس الطاقة (Entropy) لكل مجموعة بدلاً من قياس طاقة للأخطاء المتنبأة .

والطريقة الثانية تعتمد على التقليل من عدد الرموز (code) في جدول الرموز وذلك بتقسيم الأخطاء المتنبأة . وفي هذه الطريقة ، تكون الصورة باستخدام ثمانية رموز لثمانية بيت معتمداً على البيت (bit) الأخير لأخطاء التنبؤ . وإرسال البيت كما هي . ونتيجة لصغر حجم جدول الرموز ، فإن الطريقة الثانية تعالج الصور بسرعة أكبر من JPEG وتقلل من الفرق بين الترميز وفك الرمز مرة ثانية وهذا يجعل النظام متناسباً مع التطبيقات الحقيقية ولكنه يضحي بقدر من نسبة الضغط .

درجة الماجستير في العلوم
جامعة الملك فهد للبترول والمعادن
الظهران - المملكه العربيه السعوديه
شعبان ١٤١٩هـ

CHAPTER 1

INTRODUCTION

1.1 Introduction

Spectacular advances in technology over the past decade in many aspects of digital technology have altered the related disciplines and brought many applications to the area of digital image processing. The majority of modern business and consumer demand of photographs and other types of images are still processed through the more traditional analog means rather than digital means.

The key obstacle for many applications of digital image processing such as acquisition, data storage, transmission, printing and display, is the vast amount of data required to represent a digital image directly. A digitized version of a single color picture at TV resolution contains on the order of one million bytes. The use of digital images is often not viable

due to the high storage or transmission costs involved, even when image capture and display devices are quite affordable.

There has been an increased level of research in image compression[4,10,46]. Most of the efforts, however, have focused on the development of lossy compression techniques. The objective of lossy compression schemes is to obtain higher compression by sacrificing the reproduction of the exact numerical representation without introducing noticeable degradation in the reconstructed data.

Certain applications, such as medical imaging, image archiving, remote sensing, telemetry and non-destructive testing and others require lossless compression. The trend in medical imaging is increasingly digital. The basic motivation is to represent medical images in digital form to support image transfer and archiving and to manipulate visual diagnostic information in more useful and novel ways. Another push is from the PACS (Picture Archiving and Communication Systems) community [4] who envisions an all-digital radiological environment in hospitals for acquisition, storage, transmission and display of large volume of images. About 30% of radiological examinations in the United States are taken directly through digital media. The amount of digital radiological images captured per year in the United States alone is in the order of quadrillion bytes or 10^{15} and is increasing every year [4]. For a FDDI (Fiber Distributed Data Interchange) network of 100 megabits per

sec(Mb/s) and at an optimistic 20% actual transfer rate (i.e., 20Mb/s), it would take 13 seconds to transmit a digitized chest image of 4096 pixels by 4096 pixels of 12-bits resolution; where many medical applications demand the display of image in less than 2 seconds [4]. This transmission problem is further aggravated in WAN (Wide Area Network) applications which often include low-bandwidth channels, such as long distance telephone lines or an ISDN (Integrated Services Digital Network) of data rate 144 kb/s.

It is thus apparent that in order to transmit, store or display digital images, some form of compression is necessary. Fortunately, images of daily life usually have enough inherent spatial redundancy. Modern image compression technology offers a possible solution by taking an advantage of this redundancy.

The goal of the image compression algorithm is to represent the image data with as few bits as possible while simultaneously attempting to preserve the original fidelity. This compression can be divided into lossless and lossy compression. The ultimate receiver of lossy compression is the human eye. Lossy compression increases compression ratio by permitting distortion in the reconstructed data. A lossless scheme achieves modest compression ratios of the order of two but lossy compression can achieve much higher compression ratio

ranging from a ratio of 10:1 to a ratio of 50:1 or more. Higher compression ratio is achieved at the expense of more image degradation. Lossless compression permits exact bit for bit reconstruction of original data after decompression. Lossless coding techniques have been applied in three primary application areas: text compression, binary image compression and signal compression.

Although the techniques employed in these applications differ greatly, they are all fundamentally rooted in the concept of entropy and Shannon's noiseless coding theorem.

The entropy H(x) of a discrete random variable X with a PMF (probability mass function) p(x) is defined by [21]

$$H(x) = -\sum_{x \in X} p(x) \log_2 p(x) \tag{1.1}$$

The entropy is expressed in bits/sample.

The noiseless coding theorem guarantees that as long as the average number of bits per source symbol at the output of the coder exceeds the entropy, the data may be decoded without error.

The expected length L(C) of a source code C(x) for a random variable X with probability mass function p(x) is given by

$$L(C) = \sum_{x \in X} p(x) l(x) \tag{1.2}$$

Where l(x) is the length of the codeword associated with x.

Lossless coding techniques for signal data almost always consists of two stages: Decorrelation and entropy coding. The basic block diagram of lossless compression is shown in Figure 1.1. Decorrelation can be accomplished by many techniques [10,14,15,20,26,45] including prediction, transform and multiresolution techniques. Once the data has been decorrelated and if the probability mass function of the resulting samples is not uniform, compression can be achieved by entropy coding.

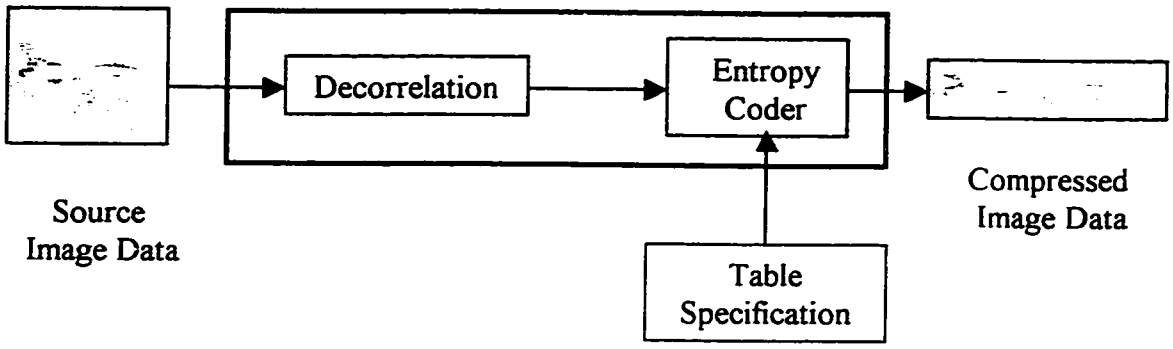


Figure 1.1: Basic block diagram of lossless encoder

One notable technique, which does not follow the two-stage model, is bit-plane encoding. Bit-plane coding is one of the earliest techniques employed for lossless coding of gray-scale images. Bit-plane coding [26,31,34,35] splits the original image into a collection of binary images, each of which is constructed by collecting bits of equal binary weight

from each pixel. Each bit-plane is then separately compressed using run length coding. As the resolution of the data increases, the performance of bit-plane decreases, since the least significant bits of each pixel do not exhibit long runs.

In this thesis two new schemes are developed. The first scheme addresses the reduction of bit rate and the second scheme addresses the real time applications. A review of classical and contemporary literature and international standards pertaining to lossless compression is presented in chapter 2. Description of the design, implementation and experimental results for the first scheme are given in chapter 3. Chapter 4 focuses in the second scheme and its results. Chapter 5 concludes the thesis and gives recommendation for future direction of research in this area.

CHAPTER 2

LOSSLESS IMAGE COMPRESSION : A REVIEW

2.1 Introduction

Like most technological breakthroughs, data compression techniques are based on theoretical research spanning decades. A lossless compression technique consists of two main components, modeling and encoding. A model captures the structure inherent in the raw data and extracts it. The residual, also called the error, is then encoded using an entropy-encoding technique. Encoding techniques such as arithmetic and Huffman encoding, are known to perform optimally in terms of the number of bits used to encode a given data source. Hence statistical modeling of the source being compressed plays a central role in any data compression system. Fitting a given source well with a statistical model is an ongoing and difficult academic endeavor pursued by researchers in a wide range of disciplines.

In lossless image compression, the task of modeling is usually split into two stages. In the first stage, a prediction model is used to predict the pixel values; and in the second stage pixel values are replaced by the errors. There has been a considerable work carried out to find the prediction models. Predictive techniques, transform techniques and multiresolution techniques are different examples of decorrelation technique.

2.2 Predictive Techniques

Predictive coding is simple to apply and it performs as well as other techniques [11]. Linear prediction [13], nonlinear prediction [26], adaptive prediction [15], frame to frame prediction [30], prediction tree [19] are examined to find the optimum prediction paths for image data. Linear prediction is frequently referred to as DPCM (Differential Pulse Code Modulation) in lossless coding. DPCM is a popular and a widely used prediction method for lossless compression [11].

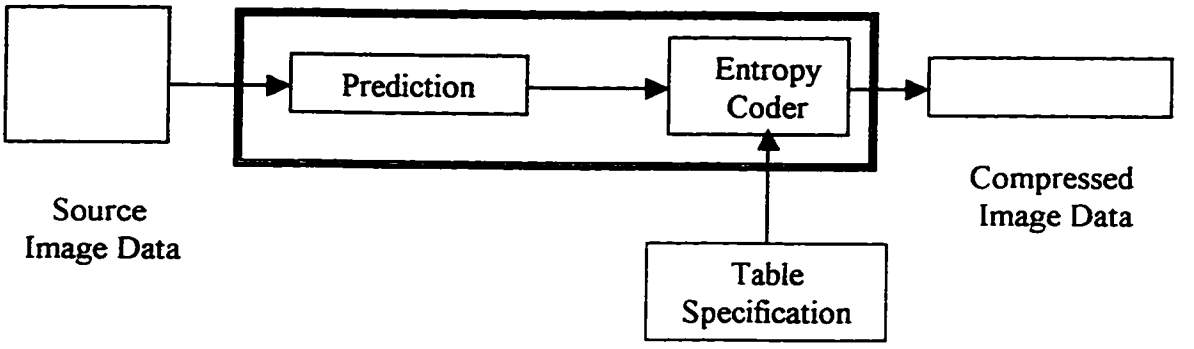


Figure 2.1: Block diagram of DPCM based lossless encoder

In DPCM, an image is typically encoded one pixel at a time across a raster scan line from left to right for two consecutive raster lines. The block diagram of a DPCM based lossless encoder is given in Figure 2.1. The value of a pixel is predicted as a linear combination of a few neighboring pixel values that have been previously reconstructed [4]. The predicted value of a pixel, $x_{est}(m,n)$ is

$$x_{est}(m,n) = \sum_{(i,j)\in ROS} \sum \alpha(i,j) x_r(m-i,n-j) \qquad (2.1)$$

where $\alpha(i,j)$ is known as the prediction coefficient or the weighting factor obtained by some optimization technique. For the purpose of coding, the ROS (Region Of Summation or Support) is chosen such that the set $\{x_r(i,j)\}$ includes only those pixels whose values have been already

decoded or reconstructed at the receiver. For example, the predicted value of the pixel x(m,n) can be expressed using three adjacent neighbors as:

$$x_{est}(m,n) = \alpha(0,1)\, x_r(m,n-1) + \alpha(1,0)\, x_r(m-1,n) + \alpha(1,1)\, x_r(m-1,n-1) \qquad (2.2)$$

The prediction error e(m,n) which is then entropy coded is given by

$$e(m,n) = x(m,n) - x_{est}(m,n) \qquad (2.3)$$

Many researchers have focused on developing a best prediction model. Nasir D. Memon et al. investigated the performance of JPEG (Joint Pictures Expert Group) linear predictor on RGB color images [29]. Giridhar Mandyam et al. investigated finding the optimal coefficients for two dimensional linear predictors [13]. Paul G. Howard presented a simplified model, which can couple the prediction and the error modeling on the fly [45]. This method is also intended for fast encoding and decoding. Xiaolin and Nasir Memon used non linear predictors which can be adapted via error feedback mechanism [26]. McCoy et al. used reversible filter as an adaptive predictor and stated sufficient condition for lossless filtering [14]. Martucci initially presented a median adaptive predictor [8] but Weinberger presented an MED (Median Edge Detector) latter which selects the median of a set of three predictors to predict the

current pixel [8]. One way of interpreting MED predictor is that it always choses either the best or the second best predictor among the three predictors. Slaven Marusic and Guang Deng has presented two adaptive predictors [15]. Their first scheme is derived from weighted median filter theory and extends to the median adaptive prediction technique which was presented by Martucci [8], and the second technique is a context based modeling technique which determines the optimum prediction by establishing the dominance of a particular predictor for each image context. B. Gandhi et al. used a predictor that was adapted to horizontal and vertical gradients in the neighborhood of the pixels being predicted [8].

By their predictor, the current pixel is predicted to be

$$\hat{x} = \alpha.A + (1-\alpha).C \quad ,$$

A	C
B	x

$$\alpha = \frac{\Delta_v}{\Delta_h + \Delta_v} \qquad \Delta_v = |A-B| \qquad \Delta_h = |A-C|$$

where x is the current pixel value, $\hat{x}$ is the predicted pixel value, and A,B,C are the neighboring pixels.

X. Wu et al. proposed a GAP (Gradient Adjusted Predictor) which adapts the prediction according to local gradients, and hence give a more robust performance compared to standard linear predictors [8]. GAP weighs the neighboring pixels according to the estimated gradients. In GAP the gradient of the intensity function at the current pixel $p[i,j]$ is estimated by computing the following quantities:

$$d_h = |\, p[i,j-1] - p[i,j-2] \,| + |\, p[i-1,j] - p[i-1,j-1] \,| + |\, p[i-1,j+1] - p[i-1,j] \,|$$

$$d_v = |\, p[i,j-1] - p[i-1,j-1] \,| + |\, p[i-1,j] - p[i-2,j] \,| + |\, p[i-1,j+1] - p[i-2,j+1] \,|$$

A prediction $\hat{p}[i,j]$ is then obtained by the following procedure

$IF\ ((d_v - d_h) > 80)\ \{sharp\ horizontal\ edge\}$
  $\hat{p}[i,j] = w$
$ELSE\ IF\ ((d_v - d_h) < -80)\ \{sharp\ vertical\ edge\}$
  $\hat{p}[i,j] = n$
$ELSE\ \{$
  $\hat{p}[i,j] = (w+n)/2 + (ne - nw)/4;$
  $IF\ ((d_v - d_h) > 32)\ \{horizontal\ edge\}$
    $\hat{p}[i,j] = (\hat{p}[i,j] + w)/2$
  $ELSE\ IF\ ((d_v - d_h) > 8)\ \{weak\ horizontal\ edge\}$
    $\hat{p}[i,j] = (3\hat{p}[i,j] + w)/4$
  $ELSE\ IF\ ((d_v - d_h) < -32)\ \{vertical\ edge\}$
    $\hat{p}[i,j] = (\hat{p}[i,j] + n)/2$
  $ELSE\ IF\ ((d_v - d_h) < -8)\ \{weak\ vertical\ edge\}$
    $\hat{p}[i,j] = (3\hat{p}[i,j] + n)/4\ )\ \}$

where

$n = p[i, j-1]$, $w = p[i-1, j]$, $ne = p[i+1, j-1]$, $nw = p[i-1, j-1]$, $nn = p[i, j-2]$, $ww = p[i-2, j]$.

Don Speck proposed another predictor that used a weighted combination of 5 neighborhood pixels in order to predict the current pixels [8]. The weights are adapted on the fly as encoding progresses. Ueno and F. Ono Clara used a predictive scheme that adaptively switches between a fixed set of predictors based on texture and gradients in the neighborhood of the target pixel [8]. Nasir Memon and Khalid Sayood investigated frame to frame prediction for lossless compression of video sequences [30]. In another work Nasir Memon et al. divided the image into blocks and found the best scanning pattern or prediction tree [19] for each block to obtain the minimum entropy of the prediction errors. This method needs some overhead for every block to encode the information about scanning pattern by which the block is scan. The possible scanning patterns are stored and given an index for each pattern. This method combines prediction scheme with effective error modeling. The analysis of common scanning techniques for lossless image coding is given in reference [20]. Some of the features on prediction error coding has been studied by Glen G. Langdon and Bryan Mealy [18]. M. Das and J. Anand proposed a lossless compression scheme based on 2-D predictive model [17]. Authors have combined two

dimensional predictive image model with a new contextual Huffman coder.

DPCM is simple and it performs as well as more complicated scheme for many types of data [11].

2.3 Transform Techniques

Transform technique [22] is frequently employed in lossy compression. It is difficult to use for lossless coding, because its coefficients are either real-valued or complex-valued. Among the transforms DCT (Discrete Cosine Transform) is widely used.

Since the DCT was first introduced in 1974, it has been adopted in various standards for image compression. The two-dimensional FDCT (Forward Discrete Cosine Transform) of a sequence $f(x, y)$ for x, y = 0, 1, 2 ....., N-1, can be defined as

$$C(u,v) = \frac{2}{N} C(u)C(v) \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} f(x,y) [\cos(2x+1)\frac{u\pi}{2N}][\cos(2y+1)\frac{u\pi}{2N}] \qquad (2.4)$$

For $u, v = 0,1,2.....,N-1.$

The correspond*ing* IDCT(Inverse Discrete Co sin *e* Transform) is defined as

$$f(x,y)=\frac{2}{N}\sum_{x=0}^{n-1}\sum_{y=0}^{n-1}C(u)C(v)c(u,v)[\cos(2x+1)\frac{u\pi}{2N}][\cos(2y+1)\frac{u\pi}{2N}] \qquad (2.5)$$

For $x, y = 0,1,2.....,N-1$

where $C(u),C(v)=\frac{1}{\sqrt{2}}$ *for* $u,v=0$

$C(u), C(v) = 1$ *otherwise*

The optimal block size N for DCT is eight, so this block size became a standard in JPEG. In lossy compression FDCT coefficients are subject to quantization but in losslesss compression this is not acceptable, because the process must be reversible.

Transform coding can be made lossless by inverse transforming the output of the lossy coder and subtracting the result from the original data and entropy coding the difference [12]. The block diagram of Figure 2.2 shows the lossless encoder by a DCT based transformation.

For the generation of the prediction errors, transform technique applies both FDCT and IDCT in the encoder side. It also entropy codes both the quantiztion coefficients and the prediction errors. Transform coding is more complex than DPCM and it performs as good as DPCM [11] despite this complexity. One transform, which can be directly applied to lossless signal coding, is the discrete WHT (Walsh-Hadamard

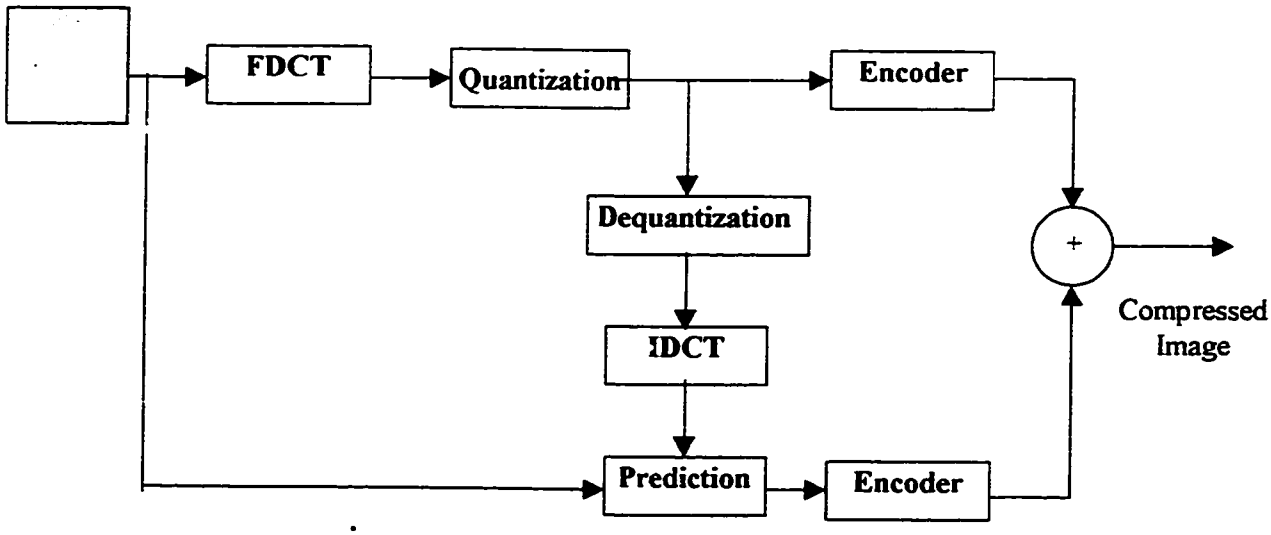


Figure 2.2: Block diagram of DCT based lossless encoder

transform). Since the coefficients of the WHT are binary fractions, quantiztion is not necessary [12].

Ho-Youl Jung et al. proposed a new types of transform which is called RT (Rounding Transform) [22]. It is an integer to integer transform. RT is reversible by virtue of a pair of the interchangeable rounding operations. A.R. Calderbank, Ingrid Daubechies et al. have investigated integer to integer wavelet transforms [23] for lossless image compression.

Due to complexity and comparable performance [11] transform coding is not used frequently in strict lossless compression.

2.4 Multi-Resolution Techniques

A number of multi-resolution techniques including HINT (Hierarchical Interpolation) [4], the laplacian pyramid [28] and the S-transform have been successfully employed in the decorrelation of the image data. These methods present the original data with varying resolutions. Therefore, these techniques also support a progressive transmission which allows data to be decoded in several stages. HINT starts with a subsampled version of the original data set by taking alternate rows and columns.

In Hierarchical Interpolation [4], shown in Figure 2.3, a sub-sampled version of the original image Δ is transmitted initially using DPCM. In the second stage, the intermediate pixels (O) are estimated by a linear interpolation of the four surrounding Δ pixels. This estimate is first rounded to the nearest integer to ensure reversibility and then subtracted from the actual pixel value. The difference which is usually very small is coded by VLC (variable length code). Now the X pixels can be estimated from Δ and O and subsequently the ♦ and ✱ pixels. The resulting transmitted image comprises the ensembles of error terms. The interpolation has essentially been used as a predictor.

The reconstruction process proceeds in a similar manner. First the Δ's are received. The intermediate pixels (the O's) can be recovered

exactly by interpolating four pixels of the low resolution image and correcting this estimation with the corresponding error terms obtained from the transmitted image. Analogously , the X, ♦ and ✱ pixels are recovered.

Δ	✱	X	✱	Δ	✱	X	✱	Δ
✱	♦	✱	♦	✱	♦	✱	♦	✱
X	✱	O	✱	X	✱	O	✱	X
✱	♦	✱	♦	✱	♦	✱	♦	✱
Δ	✱	X	✱	Δ	✱	X	✱	Δ
✱	♦	✱	♦	✱	♦	✱	♦	✱
X	✱	O	✱	X	✱	O	✱	X
✱	♦	✱	♦	✱	♦	✱	♦	✱
Δ	✱	X	✱	Δ	✱	X	✱	Δ

Figure 2.3: Classification of the pixels in hierchical interpolation (block size 4 x 4)

Gopinath et al. investigated the Hierarchical Interpolation and compared the performance with other methods [11]. Pyramid algorithm which is also a multiresolution technique has been investigated by Bruno et al. [28]. They modified the laplacian pyramid to suit the lossless compression.

The Hierarchical Interpolation performs comparably with DPCM and it suits progressive transmission.

2.5 Bit Plane Coding

Bit plane coding is a technique which does not follow the two stage method to decorrelate the image data. It breaks the image from multilevel to bi-level by taking the equal weight binary values. Bi-level images are coded by bi-level algorithm. The bi-level algorithm JBIG (Joint Bi-level Image expert Group) is widely used for bi-level image compression [34]. A block diagram of a progressive JBIG binary image compression encoder is given in Figure 2.4.

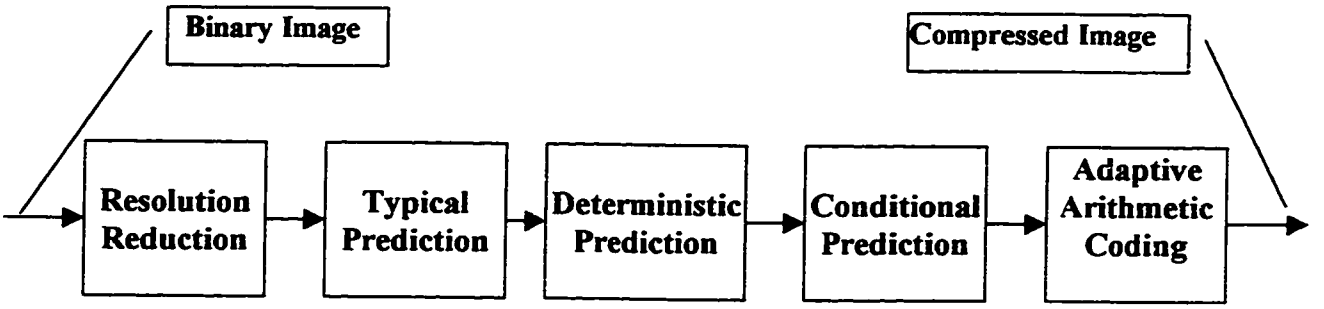


Figure 2.4: Block diagram of JBIG compression Encoder

Mourad Abdat and Maurice G. bellanger have studied JBIG algorithm for lossless compression [34]. They showed that the performance of JBIG improves when gray coding the original image before separating the bit-plane. Chowdhury et al. encodes the bit planes by their own algorithm called logic minimization [35].

Bit plane encoding does not show comparable performance with DPCM but it is useful for progressive transmission [31].

2.6 Entropy Coding

Most of the decorrelation techniques generate errors that are subject to entropy coding. Unless the PMF of the decorrelated data is uniform, entropy coding can be applied to compress the data. Most signal compression schemes employ Huffman [37] or arithmetic [38] coder. In addition several compression schemes use sub-optimal variable length coders [45] specially designed for speed or ease of implementation. The LZW (Lempel-ziv-welch) coder [44] which was developed for text compression, has also been applied to signal compression [9].

A significant difficulty in employing the above techniques is that alphabets for signal compression tend to be large, leading to implementation that require massive computational resources. Large alphabet means that the average codelength per alphabet is large and leads to increase the encoding and decoding time. To solve this problem, several approaches have been developed. One approach, which has been considerably used in image compression applications, is the Rice coder [9]. Usually coupled with a simple DPCM predictor, the Rice coder consists of several very simple coders that are nearly optimal over a

narrow range of source entropies. The system adapts to the input data by estimating the source entropy and selecting the appropriate coder. Another solution to the problem of coding large alphabets is to define a number of contexts based coder [27] based on the characteristics of the samples near the sample being coded. The context is then used to select one of several smaller coders.

Khalid Sayood and Karen Anderson proposed another approach which does not produce errors but compares the bits with the previous pixels and produces a prefix and a suffix [42]. The prefixes are encoded by small code table and the suffixes are sent directly. The maximum number of symbols in this approach is nine for eight bit pixels. This approach reduces the time required for encoding and decoding which leads to real time application. Bi-level coding [25] is also suitable for coding large alphabets.

Recently Koen Denecker et al. compared the different lossless image coder in terms of compression ratio vs encoding and decoding time [6]. They showed that speed is achieved at the expense of compression ratio.

2.7 Lossless Still Image Compression Standards

The economics of market place dictate that any product intended for communications applications must have a maximum capability of operation with similar equipment designed else where. Thus a common communications format has to be established together with the means of generating and decoding the data stream. This provides benefits in compatibility, interoperability, cheaper hardware, etc.

An impressive example of the consequences of setting up standards is the increase of usage in digital facsimile. The development of standards by the ISO (International Standard Organization), the ITU-T (International Telecommunication Union- Telecommunication Division) formerly known as CCITT (Consultative Committee of International Telegraph and Telephone) and the IEC (International Electrotechnical Commission) for audio, image and video, transmission and storage led to a worldwide activity in developing the products applicable to a number of diverse disciplines.

JPEG is an acronym for "Joint Photographic Expert Group". This group has been working under the auspices of three major international standard organizations - ISO, CCITT and IEC.

Lossless image compression algorithms rely solely on predictable characteristics of the data to be compressed. Analog data can be divided into two major types [3]:

1) Photographic data
2) Modal data

Photographic data: Analog data that predominantly yield continuous intensities (e.g. images of natural scenes) categorized in photography as "tone art".

Modal Data: Analog data that predominantly yield discontinuous intensities (e.g. images of man made documents) categorized in photography as "line art".

Analog data types are digitized with various amounts of amplitude precision, depending on the system or application requirements. The resulting digital image can also be divided into two major categories, namely:

1) Bi-level data, with only two amplitude levels.
2) Multilevel data, more than two amplitude levels.

International standards [1,2,3] for lossless compression algorithms are commonly called "Group 3" and "Group 4". "Group 3" consists of two

algorithms, MH (Modified Huffman) and MR (Modified READ (Relative Element Address Designate)). "Group 4 " consists of three algorithms, MMR (Modified Modified READ), JBIG (Joint Bi-level Image expert Group) and Lossless JPEG. They are all aimed at the subset of "still" images that require strictly lossless compression. Figure 2.5 indicates the suitability for the MH, MR, MMR, JBIG and JPEG compression algorithms.

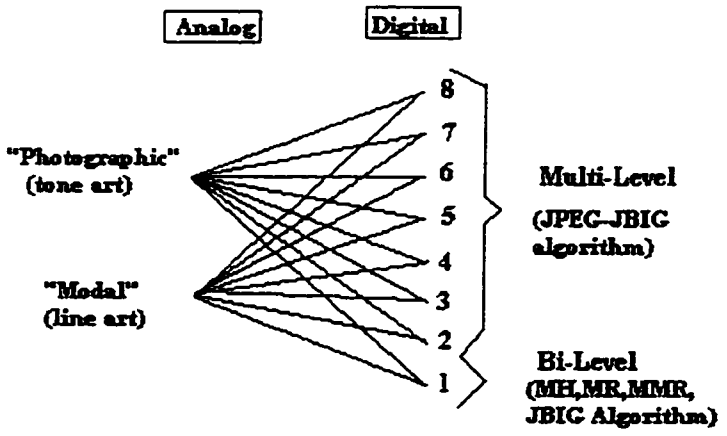


Figure 2.5: Still image data type

The MH, MR, and MMR algorithms were prepared solely by the CCITT. The JBIG and the lossless JPEG algorithms were prepared as joint effort between the CCITT and the ISO/IEC. The official designations for each algorithm are listed, along with their informal acronyms in

Table 2.1. An official standard document issued by the CCITT is called a recommendation. Such documents issued by the ISO/IEC are called international standards.

Table 2.1 – Overview of lossless algorithm

Algorithm			Compression Technology		
Algorithm Acronym	CCITT Recomm-endation	ISO/IEC International Standard	Model	Coder	Error Detection
MH	T.4	----	Static, 1D	Static Huffman	Yes
MR	T.4	----	Static 2D	Static Huffman	Yes
MMR	T.6	----	Static 2D	Static Huffman	May be
JPEG	T.81	10918-1	Custom , 3D	Custom Huffman or Adaptive arithmetic	May be
JBIG	T.82	11544	Adaptive,3D	Adaptive arithmetic	May be

2.7.1 Bi-level algorithms

Bi-level image compression algorithms used in today's digital facsimile became published standards in the early 80's. It is based on variable-to-fixed-length (run-length) coding models followed by static Huffman coding.

The names "Group 3" and "Group 4", sometimes used to refer to them, come from facsimile machine protocols that were standardized by the CCITT. It is more precise to call compression algorithms in terms of the informal acronyms MH, MR, MMR or by the CCITT recommendation T.4 or T.6 in which they are defined.

A new generation of lossless image compression algorithm, JBIG [1,34], has recently become a published standard. It uses 2D custom or adaptive modeling and adaptive arithmetic coding. It performs well over a wide range of Bi-level images including difficult digital halftone images.

The standardized algorithms for Bi-level image compression can be defined, along with the parameter settings to maximize their compression ratio as follows.

1. **MH:** Modified Huffman coding, an algorithm defined in CCITT recommendation T.4, using a 1D run-length coding model followed by a static Huffman coder. It also contains redundant end-of-line codewords to provide error detection.

2. **MR:** Modified READ (Relative Element Address Designate) coding. An algorithm defined in CCITT recommendation T.4 which uses a 2D reference and a run-length coding model followed by a static Huffman Coder.
3. **MMR**: Modified Modified READ coding. An algorithm defined in CCITT recommendation T.6, which is based on the 2D MR algorithm, modified to maximize compression by removing the MR provisions for error protection.
4. **JBIG**: Joint Bi-level Image expert Group coding. An algorithm defined in CCITT recommendation T.82, which for Bi-level compression uses an adaptive, 2D-coding model, followed by an adaptive arithmetic coder.

2.7.2 Multilevel algorithms

Two algorithms, for lossless multilevel image compression, have recently become international standards [3]. They are specific cases of the parameterizable JBIG and JPEG standards respectively.

JBIG scales to multilevel grey or color images by breaking them down into their constituent " bit-planes" and compressing these bit-planes with its Bi-level algorithms. The number of bit-planes to be compressed is specified by a parameter "P", which in the multilevel case

is 2 or more. Gray-coding [34] rather than weighted-binary coding is used to represent pixel amplitudes to maximize compression ratio when breaking an image into bit-planes.

2.7.2.1 T.82 JBIG

The JBIG standard [1] is designed mainly for progressive bi-level image coding. The image is encoded at different resolution from higher to lower resolution.

A JBIG encoder is shown in Figure 2.6. A JBIG encoder is a chain of D differential layer encoders followed by a bottom layer encoder. The case corresponding to D=0 is called sequential encoding since the image is encoded once at full resolution. Each encoding layer contains a resolution reduction module, a set of blocks for pixel prediction and context calculation and an adaptive arithmatic coder. The resolution reduction block accepts high resolution images and creates low resolution images with possibly half as many rows and half as many columns as high resolution images. The differential layer typical prediction block primary purpose is to speed up the implementation by skipping other block operations for pixels which are in a solid color region. The deterministic prediction block flags pixels which are deterministically predictable in order to avoid their coding by the adative

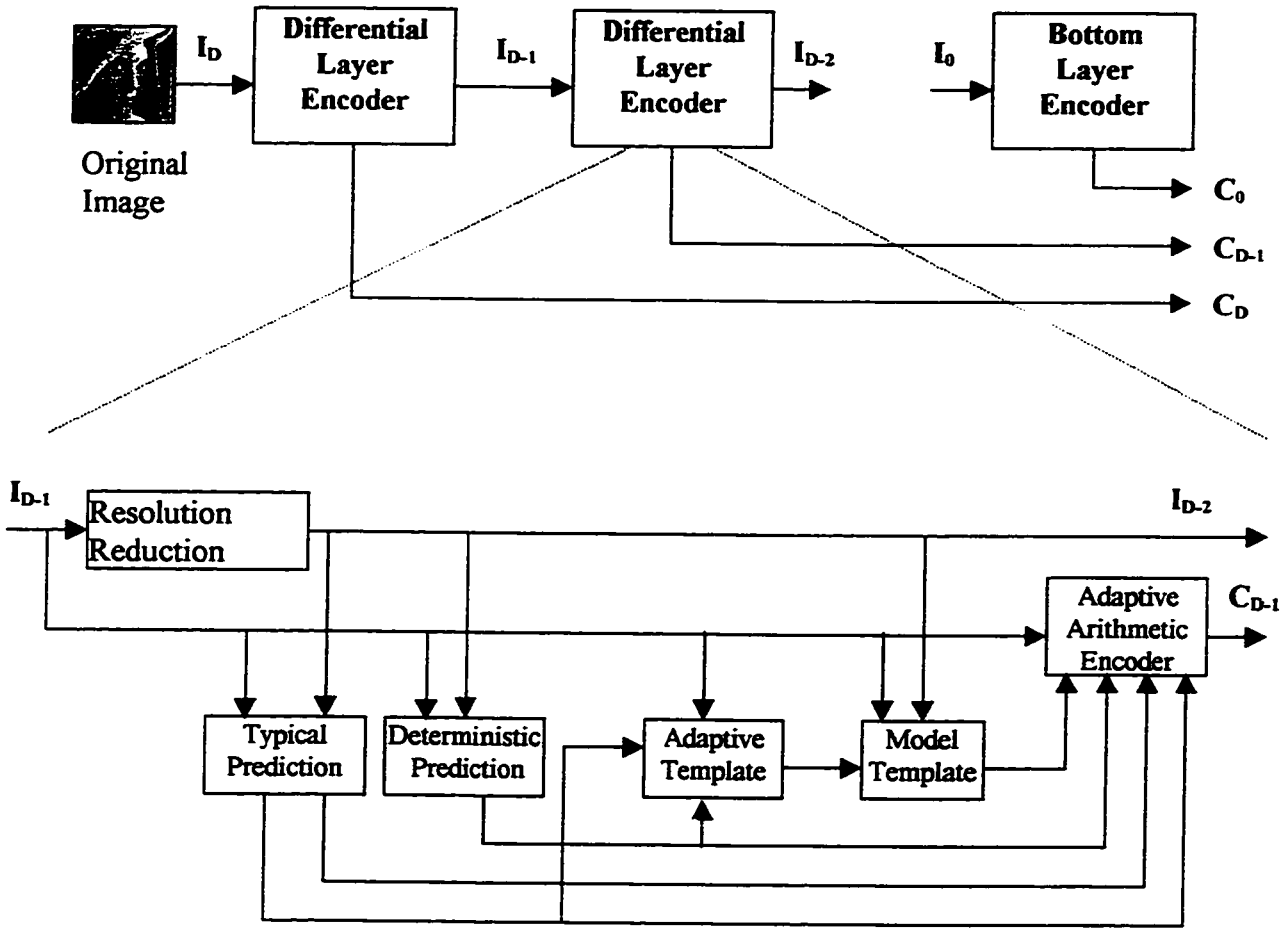


Figure 2.6: Structure of a JBIG encoder

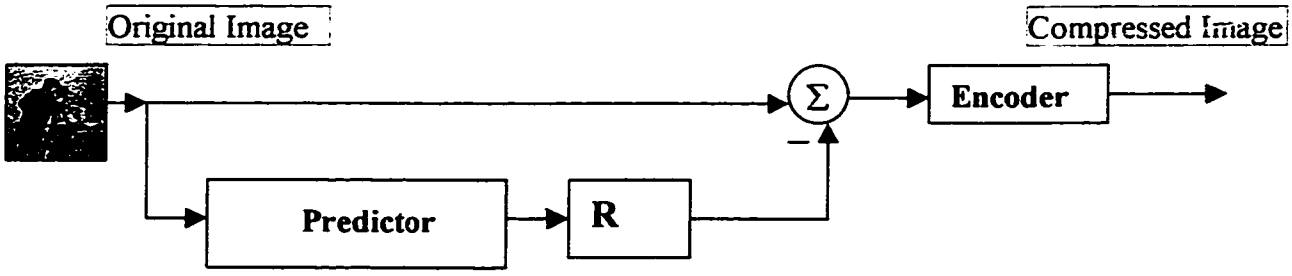
arithmatic encoder. A pixel is deterministically predictable when it is deductible from the already known pixels in the low-resolution images and those in the casual high-resolution image. The model template looks for the neighborhood pixels of the current pixel. The values of the surrounding pixels put in a particular order form an integer which is

called the context and are exploited by the adaptive arithmatic coder to code the current pixel. The adaptive template block adjusts the context of the model template block when some horizontal periodicity in the image is found. Finally the adaptive arithmatic encoder is an entropy coder which first notes the outputs of the typical prediction and the deterministic prediction blocks to determine weather it is necessary to encode a given pixel. Assuming it is necessary to encode the pixel value, then it notes the context and uses its internal probability estimator to estimate the conditional probability that the current pixel will be a given color. Often the pixel is highly predictable from the context so that the probability is very close to one 0 or 1 and a large entropy coding gain can be realized. The bottom layer encoder is less complicated than the differential layer encoder since there is neither a resolution reduction module nor a deterministic prediction block.

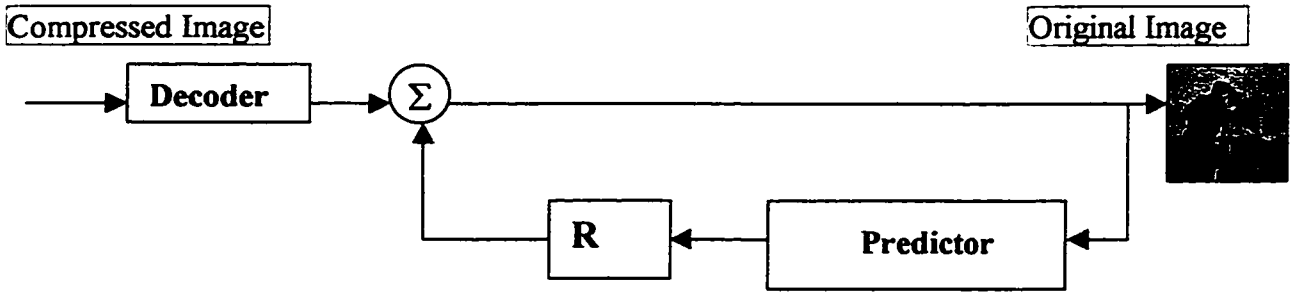
2.7.2.2 T.81 JPEG

After its selection of a DCT (Discrete Cosine Transform)-based method in 1988, JPEG discovered that a DCT-based lossless mode was difficult to define as a practical standard against which encoders and decoders could be independently implemented without placing severe constraints on both encoder and decoder implementations [2]. JPEG has chosen a

simple predictive method that is wholly independent of the DCT processing to meet its requirement for a lossless mode of operation. JPEG lossless method produces results, which in light of its simplicity, are surprisingly close to the state of the art for lossless continuous-tone compression [2]. Figure 2.7 shows the main processing steps for a single-component image. The block 'R' in this figure represents a rounding operation.



(a) JPEG encoder



(b) JPEG decoder

Figure 2.7: Block diagram of (a) JPEG encoder (b) JPEG decoder

A JPEG predictor combines the values of up to three neighboring samples (A, B and C) to form a prediction of the sample indicated by X in Figure 2.8.

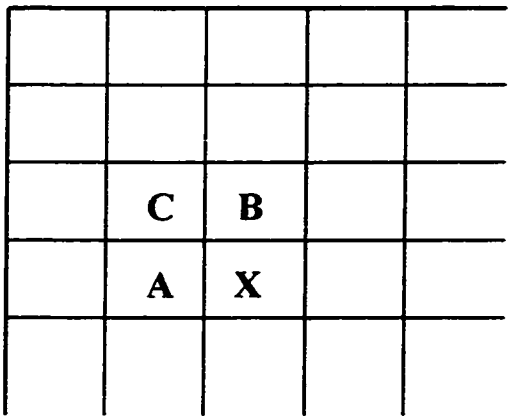


Figure 2.8: Relationship --sample and prediction neighborhood

The prediction is calculated by using any of the eight predictors shown in Table 2.2.

Table 2.2—JPEG predictors

Selection-value	Prediction
0	no prediction
1	A
2	B
3	C
4	A+B-C
5	A+(B-C)/2
6	B+(A-C)/2
7	(A+B)/2

This prediction is then subtracted from the actual value of the sample X and the difference is encoded losslessly by either entropy coding methods - Huffman or arithmetic. Selection 1, 2, and 3 are used for one dimensional predictors. Selection 4,5,6 and 7 are used for two dimensional predictors. Selection value '0' can only be used for differential coding of hierarchical mode of operation.

For lossless mode of operation, two different codecs are specified – one for each entropy coding method. The encoders can use any source image precision ranging from 2 to 16 bits/sample and can use any of the predictors except a selection of value 0. The decoder must handle any of the sample precisions and any of the predictors.

Lossless image compression varies significantly depending on many factors including the specific image and its type (Photographic or modal), as well as how it was digitized (its spatial resolution and amplitude precision).

In the case of still image compression, JPEG (Joint Picture Experts Group) has emerged as a standard with the following modes of operation [2] :

1) Sequential encoding: each image component is encoded in a single left-to-right, top-to-bottom scan.

2) Progressive encoding: the image is encoded in multiple scans for applications in which transmission time is long and the viewer prefers to watch the image buildup in multiple coarse-to-clear passes.
3) Lossless encoding: the image is encoded to guarantee exact recovery of every source image sample value (even though the result is low compression compared to lossy modes.
4) Hierarchical encoding: the image is encoded at multiple resolutions so that lower-resolution version may be accessed without first having to decompress the image at its full resolution.

For each mode, one or more distinct codecs (encoder/decoder) are specified. Codecs within a mode differ according to the precision of source image samples they can handle or the entropy coding method they use. Although the word codec used, there is no requirement that implementations must include both an encoder and decoder. Many applications will have systems or devices that require only one or the other.

The international standards organization (ISO) recently solicited proposals for a new lossless/nearly lossless compression standard for

continuous tone images. A total of nine proposals were submitted in the summer of 1995. Seven of nine proposals used a prediction step for 'decorrelating' the image prior to modeling and encoding [8]. So in lossless compression predictive coding is widely used.

In this thesis prediction step used as a mean of decorrelation to generate prediction errors and used predictors recommended by lossless JPEG standard algorithm and Huffman coding as entropy coder.

2.8 Statement Of The Problem

From the review of lossless image compression in this chapter it is apparent that efficient decorrelation of the image is mainly carried out by the prediction method. A new prediction model is intended to find the best predictor to produce the lowest entropy. But no attempt has been made to model the errors after their generation. The first part of this thesis will attempt to model the prediction errors after their generation. Modeling is done by mapping the prediction errors to a bit stream of many layers and each layer corresponds to certain prediction error. In this method, bit streams are entropy coded instead of entropy coding the prediction error.

For real time application, encoding and decoding time should be comparable. A technique is called symmetric when the difference

between the encoding and decoding time is small. Most of the prediction based image compression techniques are asymmetric and less attempt has been made to find a symmetric technique [6,9,12,44,45]. Part of this thesis will attempt to make a symmetric technique intended for real time applications. In this method, a short code table consists of only eight codes are made by grouping the prediction errors. Due to only eight codes in the code table decoding time is greatly reduced and the difference between the encoding and decoding time becomes small.

CHAPTER 3

METHOD-I: LOSSLESS IMAGE COMPRESSION BY ERROR MODELING

3.1 Introduction

Method-I is described in the first part of this chapter in details and simulation results are presented in the second part. The entropy of the prediction errors is a major obstacle that defines the lower bound of the encoded bit rate. Method-I will map the prediction errors to a bit stream. Bit streams are then encoded using a run only or a run length method for compression.

Most of the entropy reduction in Lossless Image Compression is caused by the decorrelation of the DPCM technique. Although pixels of the error image are decorrelated, the remaining small correlation between errors can be further exploited by modeling the errors. Most of the error modeling techniques are embedded within the DPCM technique

[16,19,45]. The objective of all DPCM techniques and their adaptive version for error modeling is to produce low entropy prediction errors. The prediction errors are then entropy coded by a Huffman or an arithmetic coder. The thesis work will model the errors after their generation in such a way that errors can be mapped into layers of binary numbers. Due to the large number of zeros in the layers, bit streams can be coded efficiently by run-length coding. Treating the errors in this way enables us to further decrease the bit rate. The advantage of this method is its offline use to increase compression ratio and the ability of coupling it with any prediction method. Latter in this chapter the block diagram of the proposed method and the method of forming bit stream from prediction errors and the different ways of getting statistics for encoding bit streams are described. This chapter also contains information about the overhead needed to decode images. Simulation results and the performance of Method-I are presented at the end of this chapter.

3.2 Modeling Of The Prediction Errors

Method-I of this thesis is intended to reduce the compression ratio further by introducing an additional block in the compression process. This block which is called 'error modeling' will be placed in the middle of the two stages.

The block diagram of Method-I is shown in Figure 3.1. The first block of Figure 3.1 is called 'prediction' block. In this block prediction errors are generated using JPEG predictors.

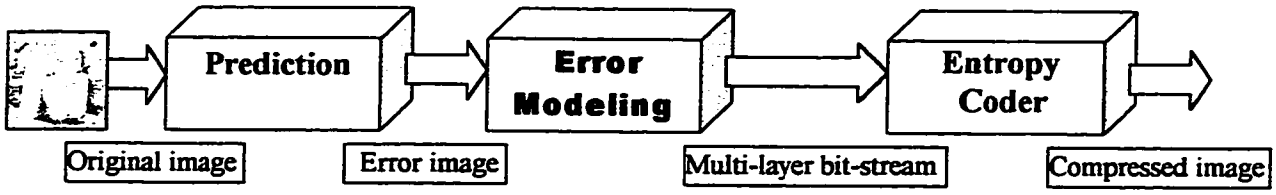


Figure 3.1: Block diagram of compression process by Method-I

Information about JPEG predictors is presented in Table 2.2 and their position relative to the current pixels has been shown in Figure 2.8 of chapter 2. The following procedure is adopted to calculate the prediction errors of the current pixel. First, calculate the predicted value of the current pixel from its neighboring pixels using a JPEG predictor. Second, subtract the predicted value from the current pixel value. The predicted values must be rounded to the nearest integer to make sure that the prediction errors have integer values. The prediction errors can be calculated by the following equation:

$$\text{Prediction error} = \text{current pixel value} - [\text{predicted value}] \qquad (3.1)$$

where the parenthesis [ ] means rounding operation.

The 'error modeling' block of Figure 3.1 will model the prediction errors. The input to this block is the prediction errors generated by the 'prediction' block and output is the bit-stream of many layers.

The histogram of the prediction errors of Lena image using two JPEG predictors shown in Figure 3.2. The histogram of the other test images almost shows the same pattern. From the observation it can be conclude that these histogram almost follow a Gausian or Laplacian function. The possible maximum number of prediction errors is 511 and the value of these prediction errors lies within the range of –255 to 255. The prediction errors can be arranged in Gausian form or can be arranged in descending probability. Method-I in this thesis will form a layer of bit stream for every prediction error present in the error image. The layers are concatenated in the same way as the prediction errors are arranged. The following procedure will describe how bit streams are generated from prediction errors.

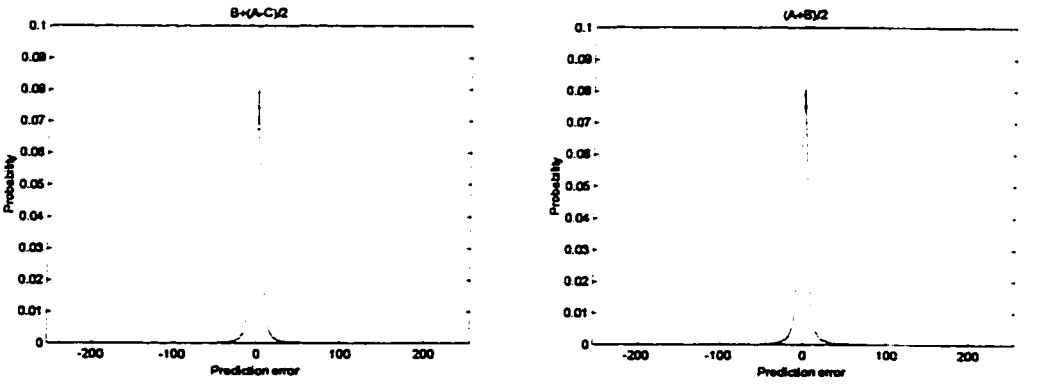


Figure 3.2: Histogram of Lena image using JPEG predictors 6 and 7

3.2.1 Generation of bit stream from error image

Figure 3.3 provides the flow chart for bit stream generation from error image. According to the flow chart, the following steps are carried out:

1. Make a list of the available prediction errors or symbols from the error image.
2. Every prediction error presents in the error image will make a layer of bit stream and the prediction errors are arranged according to the following sequence:

   0 1 −1 2 −2 3 −3 4 −4 ..... 254 −254 255 −255

   and their corresponding probabilities are:

   p_0 p_1 p_{-1} p_2 p_{-2} p_3 p_{-3} p_4 p_{-4} ..... p_{254} p_{-254} p_{255} p_{-255}

   respectively. If any of the prediction errors in the above sequence is not present in the error image then it can be skipped from the error modeling. The above sequence usually follows the highest to the lowest frequency of occurrence with small variation compared to the prediction errors arranged in descending probability.
3. Pick up the first prediction error from step 2 and start scanning the error image from the first to the last element in a raster order and compare every element with the prediction error that has been picked up. Insert '1' for that position where a match is found otherwise insert '0'. These 0's and 1's are placed in a different part

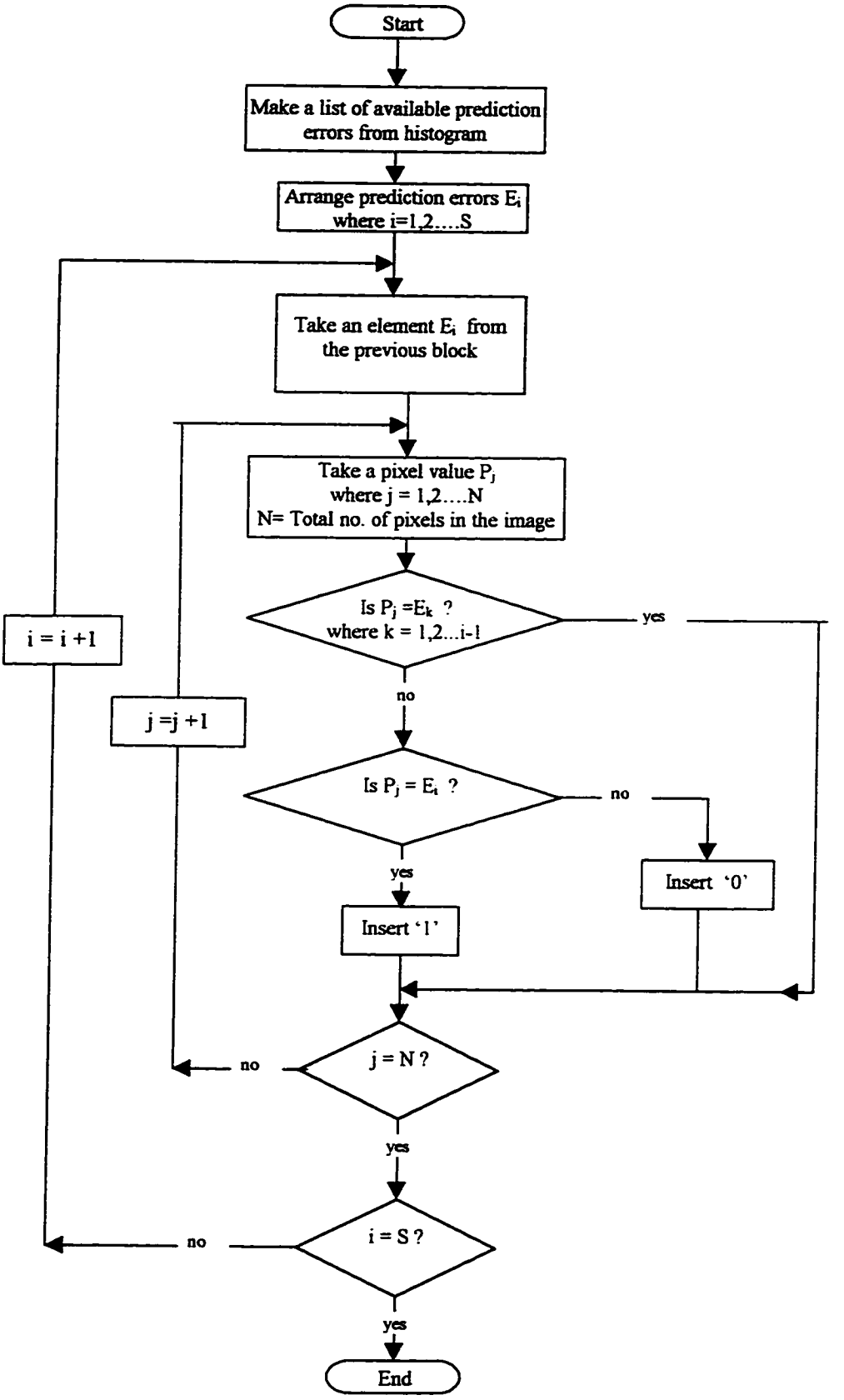


Figure 3.3: Flow chart for bit stream generation

of memory without disturbing the error image. Here, element means pixel values in the error image. Once the scanning is carried out, it will be the first layer with 0's and 1's for the first prediction error of step 2. Every '1' in this layer corresponds to the place where the first prediction error exists and '0' for other than the first prediction error.

Pick up the second prediction error from the list of step 2 and start scanning the error image from the first to the last element in the error image to form the second layer. Compare every element except the places where the first prediction error exists. Insert '1' if a match is found, if not insert '0'. This will form the second layer with 0's and 1's. These 0's and 1's of the second layer will be placed after the first layer. The total number of ones and zeros in the second layer

$$= (\textit{No. of ones and zeros in the first layer}) - (\textit{no. of ones in the first layer}) \quad (3.2)$$
$$= N - \textit{No of ones in the first layer}.$$
where N is the total no pixels in the image

The third prediction error is picked up from step 2 to form the third layer and scanning the error image is started again from the first to the last element. In this stage, element positions will not be scanned for which already '1's are present in the previous

layers. Compare elements in the error image with the prediction error that has been picked up from step 2. Insert '1' if a match is found, otherwise insert '0'.

By following this procedure, pick up the prediction error one by one from the list of step 2 and form a layer for that prediction error until the end of the list of step 2 is reached. All layers are concatenated one after another to form a bit stream.

4 The number of bits in the next layer will be known after completion of the preceding layer. The total number of 1's in the bit stream is the same as the total number of pixels in the image.

In the 'error modeling' block of Figure 3.1, the error image is converted to a multi-layer bit stream. Every layer in this bit stream corresponds to a certain prediction error and a '1' in that layer represents the position of that prediction error. In this stage, the image can be expanded but the nature of the bit-stream is such that the ratio of the number-of-zeros to the number-of-ones in the layers follow a pattern that can be compressed efficiently using a run length coding.

The total number of layers, L, in the bit stream will be;

$$L = S - 1 \tag{3.3}$$

where S is the total number of prediction errors (symbol) present in the error image.

The total number of 0's and 1's in the first layer will be equal to the total number of pixels in the image. The number of ones in the layer will be p_0Xn, where N is the total number of pixels in the image and p_0 is the probability of the first prediction error. So, the number of ones and zeros in the second layer will be the total number of bits in the first layer minus the number of ones in the first layer. So, whenever any layer is known, the number of bits in the next layer can be found. In general, the number of bits in a layer is equal to the N minus number of 1's in all previous layers. The total number of bits B in the bit stream in terms of the probability of the prediction errors will be

$$B=[1+\{1-p_0\}+\{1-p_0-p_1\}+\{1-p_0-p_1-p_{-1}\}+.....+\{1-p_0-p_1-p_{-1}-.....-p_{255}\}\ ]*N \qquad (3.4)$$

Statistics are taken from the bit stream, once the bit streams are complete. Based on the statistics, bit streams are Huffman coded to give the desired compression.

3.2.2 An example of error modeling

A 4 pixel by 4 pixel arbitrary image and its errors image using simple predictor is shown in Figure 3.4.

25	26	26	26
23	23	24	30
22	27	27	26
24	26	29	27

Original image

25	1	0	0
-2	0	1	4
-1	5	0	-1
2	2	3	-2

Error image

Figure 3.4 An arbitrary image and its error image

The prediction errors present in this arbitrary image are listed below in Table 3.1 with their number of occurrence

Table 3.1: Statistics of an arbitrary image based on JPEG

Prediction Error	No-of-occurrence	Probability
0	4	.25
1	2	.125
-1	2	.125
2	2	.125
-2	2	.125
3	1	.0625
4	1	.0625
5	1	.0625
25	1	.0625

The number of prediction errors present in the error image is nine. So the number of layer will be (9-1)=8.

From Shannon's theory the entropy H of this source will be

H=.25log(.25)+125log(.125)+.125log(.125)+.125log(.125)+.125log(.125)+
.0625log(.0625)+.0625log(.0625)+.0625log(.0625) +.0625log(.0625).

H = 3.0 bits/pixel.

Eight layers of 0's and 1's can be form by scanning the error image and shown in Figure 3.5. Every layer corresponds to a certain prediction error. When the error image will scan in raster order then the sequence of prediction errors will be as follows:

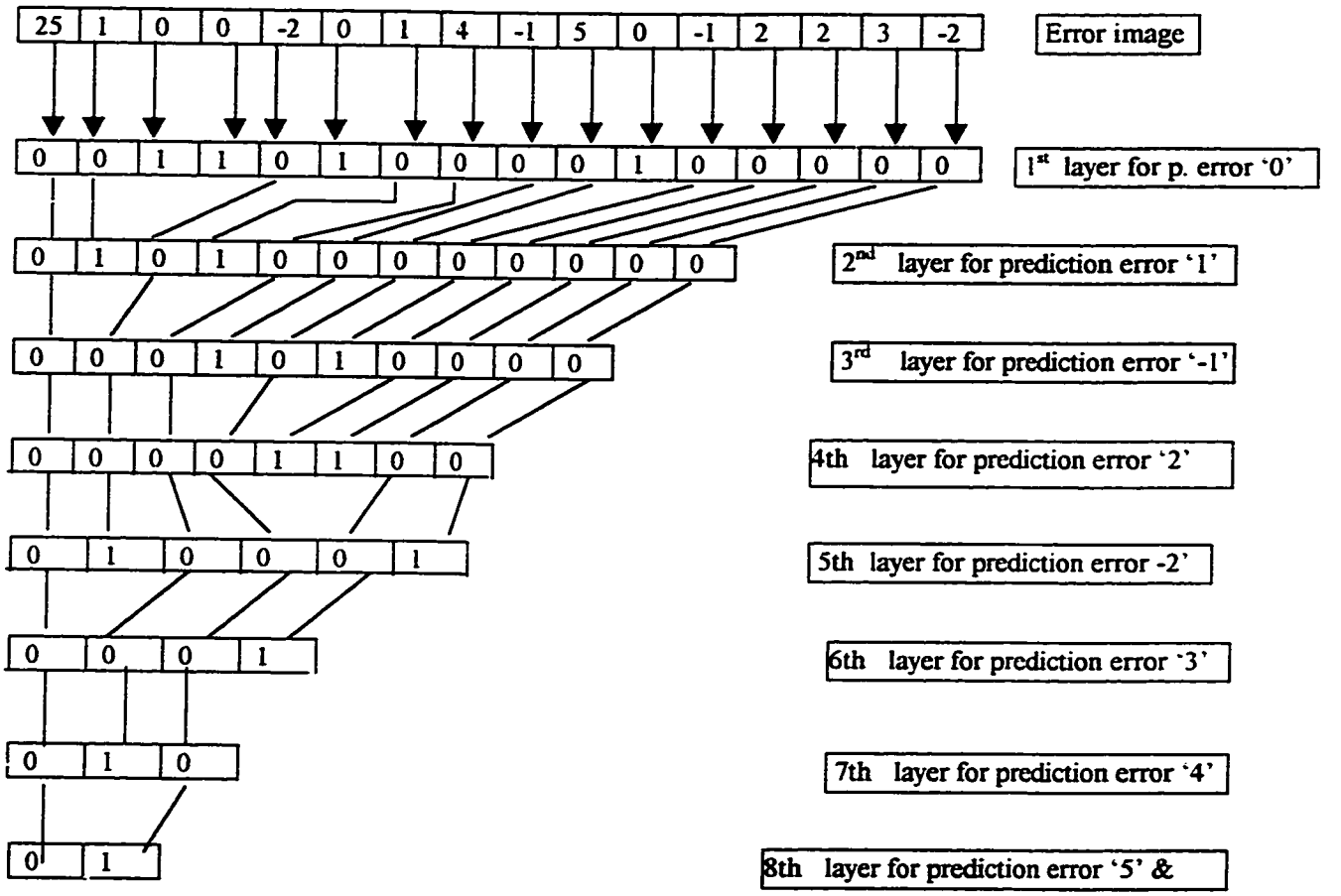


Figure 3.5 : An example of bit stream generation

The complete bit stream is shown below. The sign "||" in the bit stream shows the layer separation.

0011 0100 0010 0000 || 0101 0000 0000 || 0001 0100 00 || 0000 1100 ||

010001 || 0001 || 010 || 01 ||

Table 3.2 shows the probability of different runs when statistics are taken based on Run Only (RO). RO means counting number of zeros or runs before any single one.

Table 3.2: Statistics of run only based on Method-I

Run only (RO)	Occurence	Probability
1	4	.25
3	3	.187
0	2	.125
2	2	.125
4	1	.0625
6	1	.0625
8	1	.0625
11	1	.0625

Entropy for Run only (RO) is

H=.25log(.25)+.1875log(.1875)+.125log(.125)+.125log(.125)+.0625log(.062

5)+.0625log(.0625)+.0625log (.0625)+.0625log(.0625).

H = 2.7 bits/pixel

3.3 Encoding

A Huffman code table is generated based on the statistics from the bit streams. Bit streams generated by the 'Error Modeling' block of Figure 3.1 are Huffman encoded using the Huffman code table. In Method-I, five types of statistics are investigated. These are the sub-methods of Method-I and are as follows:

1. **Run Only (RO)**: Run only means counting number of 0's or runs before any '1'. In RO symbols are runs and length is always one.
2. **Run Length (RL)**: Run length means counting number of 0's or runs before any '1' or a stream of consecutive 1's. The number of 1's in the string is called a length. In RL, each symbols have run and length.
3. **Run Only in three groups (RO-3)**: In this case bit streams are divided into three groups. The first group is formed by taking first 25% ones, the second group is formed by the next 50% ones and the third group is formed by the last 25% ones from the bit streams. The number of groups, which is three, and the number of 1's in the different groups are chosen arbitrarily from observations. Statistics for each of the three groups are taken separately by a run only (RO).

The motivation behind this is the different concentration of ones in the different parts of the bit stream. The ratio of the number-of-zeros to the number-of-ones in the first few layers is much less than in the middle and the last layers. These ratios represent the average number of zeros for every one in that layer. Figure 3.6 shows the zero-to-one ratio curve in the different layers in a typical image Lena.

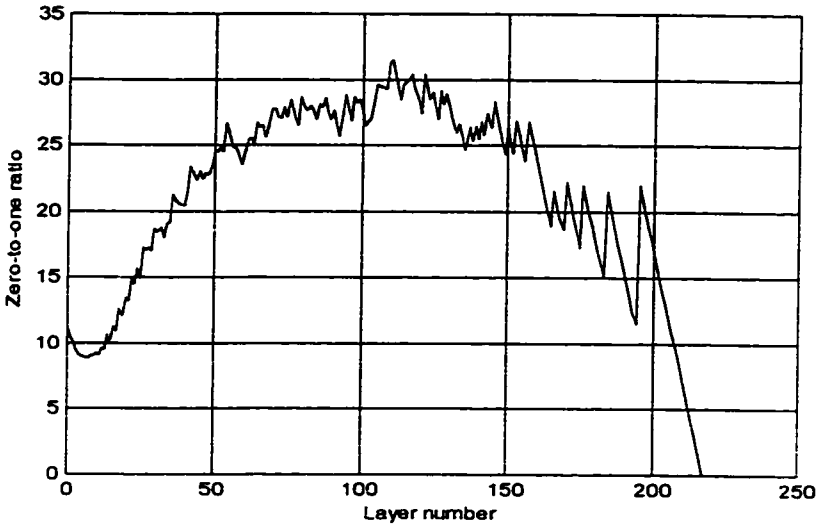


Figure 3.6: Plot of zero-to-one ratio curve in the different layers of a typical image

This pattern almost same for other images where zero to one ratio is small for first few layers and large for middle layers. The first few layers also contains the major portion of the total 1's and can be seen in Figure 3.7. The pattern of this figure is almost same as other images and shows the cumulative ones starting from the first layer to the last layer and it is a monotonically increasing function. This function is rising sharply for the first few layers and increases very slowly for the remaining layers.

It is necessary that some extra information be encoded with the encoded image by RO-3 method because bit streams are encoded using three code tables instead of one code table.

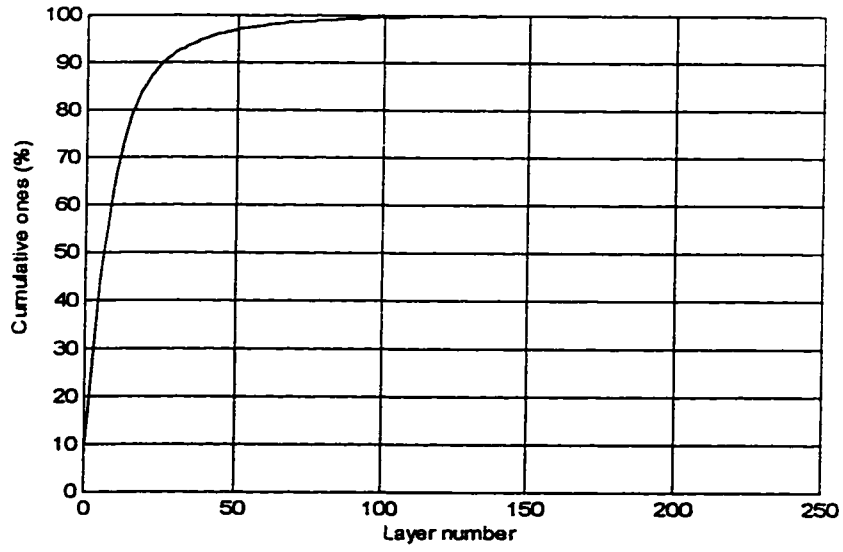


Figure 3.7: Plot of cumulative ones in the different layers of a typical image

4. **Multi-group(MG):** In this case the number of groups is not fixed. The decision on the number of groups and the number of ones in the group depends on the shape of the zero-to-one ratio curve. Figure 3.7 indicates that about 75% ones are in the first few layers and Figure 3.6 indicates the ratio of zero-to-one is less for the first few layers. From the observation of these two graphs, bit streams are divided into suitable groups. All the groups in this sub methods are encoded separately by a RO sub-method.
5. **Multi-group-mix(MG-M):** In this case, groups are the same as the multi-group(MG) but the first group is encoded by RL sub-method instead of RO sub-method. The remaining groups are encoded by a RO sub-method. The motivation is the high concentration of 1's in the first group which can be encoded efficiently by a RL sub-method instead of a RO sub-method.

3.4 Overhead

Encoding an image requires either a standard code table or a code table based on its own statistics. Standard code tables are generated from the statistics of a large number of images. All possible symbols should have a corresponding code. If statistics of an image is much different from those of a standard statistics then the bit rate will be much higher when

encode the image by standard code table. It will be efficient to encode the image based on its own code table which prevents statistical mismatch. But every compressed image must carry its own code table or enough information to generate its code table before decoding.

In Method-I, an image is encoded based on its own code table instead of a standard code table. There are two types of information that are needed to decode any compressed image by Method-I.

1. Information about each layer
2. Information about code table

The first type of overhead will tell us which layer corresponds to which prediction error. If layers, generated from prediction errors, are arranged based on the sequence described in step 2 of section 3.2.1, then the maximum number of bits needed to encode this information is 511 bits irrespective of the image size. Here every bit indicates the presence or absence of a layer. For most of the cases 300 bits are sufficient to encode this overhead. If prediction errors are arranged according to the probability, not by the sequence described in step 2 of section 3.2.1 then this overhead will not be limited to 511 bits. It will be proportional to the actual number of symbols present in the error image.

Another type of overhead is the symbol and its corresponding code. It needs three types of information – (i) symbol (ii) code length and (iii)

code, to recover any code table. But in this work instead of sending a code table, enough information is sent to generate the code table. These are symbols and their frequency of occurrence from the highest probability to the lowest probability. When arranged in descending order, frequency of occurrence is a monotonically decreasing function. Symbols in the code table also follow a pattern which can be coded with fewer bits. This overhead is proportional to the number of symbols present in the error image.

If any image is encoded by more than one code table then enough information for every code table must be encoded with the compressed image. The first type of overhead is independent of the size of the image. The second type of overhead is also independent of the size of the image but is proportional to the number of symbols present in the error image. So, if the size of the image is large, the overhead is very small compared to the encoded bits/pel and it is considerable when the image size is small.

3.5 Decoding

The block diagram of decompressing the images obtained by Method-I is shown in Figure 3.8. Like encoding, decoding also has three blocks but in the reverse order. The input to the first block is the compressed image

with overhead. This block recovers initially the information of prediction errors present in the different layers. Symbols and their frequency of

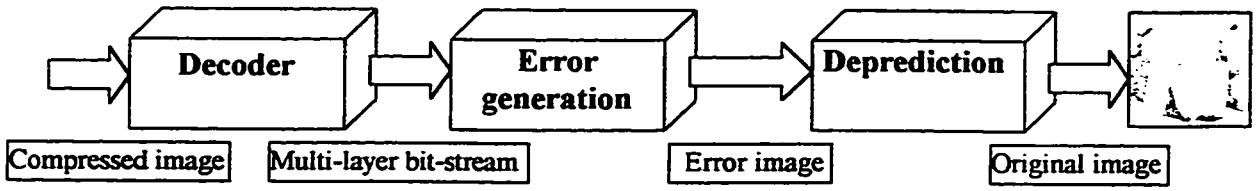


Figure 3.8: Block diagram of decompression process by Method-I

occurrence are consequently recovered. When all symbols and their occurrences are known, a code table is generated. Based on this code table the remaining bits are decoded and the output of the 'decoder' block is a multi-layer bit stream. Each layer corresponds to different prediction errors. This bit stream will be an input to the middle block called 'Error generation' and its output is the error image. Generation of an error image from the bit stream is the same as the generation of bit streams from the prediction errors but in the reverse order. The third block 'Deprediction' will predict the pixel value and will add the predicted value with the error image to recover the original pixel value.

3.6 Results

In the following section simulation results are presented for Method-I. Method-I is used to reduce the bit rate or increase the compression ratio in comparison with JPEG by modeling the errors. The images used for simulation are widely used by the image processing community. Two groups of images are used for the simulation. The first group consists of ten images from different classes. The second group consists of four images from the same class of medical X-ray images. These images are given in the appendix A and B respectively.

A summary of the results is presented for the first group in a tabular and in a graphical form. Results for the second group of images are presented for every image. Lossless image compression almost reaches saturation and improvement in the literature becomes very nominal. Achievable compression ratio varies within a few percentage in the order of two depending upon the class of the image. Because of this low compression ratio, results in the literature usually are compared in terms of bit rates (bits/pixel). At the end of this chapter complexity incurred by Method-I is discussed. Finally, results are summarized with conclusions.

3.6.1 Organization of the results

The results for the first group of images are presented in Tables 3.3(A) and Table 3.3(B). Tables 3.3(A) and 3.3(B) consists of three major columns. Every major column has four sub columns for different methods. The first major column of Table 3.3(A) presents the entropy by different methods, the second major column presents the actual bit rate achieved after Huffman coding and the third major column presents the number of symbols in the code table.

The first major column of Table 3.3(B) presents the overhead needed by different methods. These overheads are calculated by taking JPEG as reference. The second major column presents the ultimate bit rate needed including the overhead to compress the image. The differences in bit rate by different methods taking JPEG as a reference are given in the third major column. A negative difference in the third major column of Table 3.3(B) indicates the bit rate improvement.

These results are also presented in a graphical form. The graph has four points, one for JPEG, one for Method-I RO, one for Method-I RL and another for Method-I RO-3.

Results for the second group of images are presented in tables and graphs. Every image of the second group has separate tables and graphs. Results for every image are generated using different JPEG predictors.

3.6.2 Results of the first group of images

Results for the first group of images are presented in Table 3.3. For the sake of comparison the minimum bit rates are bold faced and underlined in the second major column of Table 3.3(B). From these bold faced values, the minimum bit rate is achieved for nine out of ten images by Method-I and only one image shows the minimum bit rate by JPEG. Out of the improved nine images by Method-I, four images are improved by RO-3, three images by RL and two images by RO. These results are presented in Table 3.4(A).

Comparing different sub-methods of Method-I separately with JPEG, Method-I RO shows a bit rate improvement in eight out of ten images, Method-I RL shows a bit rate improvement in nine out of ten images and Method-I RO-3 shows a rate improvement in eight out of ten images. These results are presented in Table 3.4(B). As a whole, Method-I shows a bit rate improvement over JPEG by 80% to 90% images when this algorithm is applied to images from different classes.

Comparing the different sub methods of Method-I, RO-3 performs well in five images, RL performs well in three images and RO performs well in two images. These results are summarized in Table 3.4(C). So, when Method-I is chosen for compression, RO-3 will be best choice among these three sub-methods.

Table 3.3: Summary results of Group-1 images

(A)

Image	Entropy (bits/pixel)				Bit rate after huffman coding w/o overhead				Symbol in code table			
	JPEG	Method-I			JPEG	Method-I			JPEG	Method-I		
	Lossless JPEG	RO	RL	RO-3. 25:50:25	Lossless JPEG	RO	RL	RO-3. 25:50:25	Lossless JPEG	RO	RL.	RO-3. 25:50:25
Lady	4.471311	4.44113	4.42367	4.39589	4.497330	4.47404	4.44627	4.421158	160	204	455	402
Couple	4.188632	4.12738	4.03984	4.048197	4.209585	4.15182	4.06099	4.071930	156	167	526	333
Camera	4.905333	4.89622	4.83832	4.698197	4.932617	4.92047	4.85684	4.729462	282	373	766	692
Bridge	6.007572	5.99102	5.97160	5.96753	6.040405	6.01921	5.99995	5.995132	218	241	399	608
Lena	4.933578	4.91391	4.88032	4.903630	4.951461	4.94323	4.90888	4.930840	193	249	528	522
Baboon	6.274805	6.21816	6.2195	6.21194	6.298161	6.25227	6.24776	6.240757	247	501	879	1125
Peppers	4.868655	4.86980	4.86647	4.856485	4.891953	4.89534	4.89175	4.882896	192	311	598	556
Airplane	4.323251	4.33090	4.30649	4.252542	4.356461	4.36142	4.32725	4.279873	163	258	582	485
Boats	4.476017	4.38201	4.36749	4.362935	4.505821	4.41971	4.39004	4.393813	181	355	786	644
Goldhill	5.110598	5.10617	5.04583	5.069642	5.136806	5.13323	5.07210	5.098240	326	299	777	790

(B)

Image	Overhead (bits/pixel)				Total bit rate including overhead (bits/pixel)				Difference in bit rate from JPEG (bits/pixel)			
	JPEG	Method-I			JPEG	Method-I			JPEG	Method-I		
	Lossless JPEG	Run only (RO)	Run leng. (RL)	Run in grp. (RO-3) 25:50:25	Lossless JPEG	Run only (RO)	Run leng. (RL)	Run in grp. (RO-3) 25:50:25	Lossless JPEG	Run only (RO)	Run leng. (RL)	Run in grp. (RO-3) 25:50:25
Lady	0.00000	0.00537	0.03601	0.02954	4.49733	4.47941	4.48228	**4.45070**	0.0000	-0.0179	-0.0150	-0.0466
Couple	0.00000	0.00134	0.04517	0.02161	4.20959	4.15316	4.10616	**4.09354**	0.0000	-0.0564	-0.1034	-0.1161
Camera	0.00000	0.01111	0.05908	0.05005	4.93262	4.93158	4.91592	**4.77951**	0.0000	-0.0010	-0.0167	-0.1531
Bridge	0.00000	0.00281	0.02209	0.04761	6.04041	**6.02202**	6.02204	6.04274	0.0000	-0.0184	-0.0184	0.0023
Lena	0.00000	0.00171	0.01022	0.01004	4.95146	4.94494	**4.91910**	4.94088	0.0000	-0.0065	-0.0324	-0.0106
Baboon	0.00000	0.00775	0.01929	0.02679	6.29816	**6.26002**	6.26705	6.26755	0.0000	-0.0381	-0.0311	-0.0306
Peppers	0.00000	0.00363	0.01239	0.01111	**4.89195**	4.89897	4.90414	4.89401	0.0000	0.0070	0.0122	0.0021
Airplane	0.00000	0.00290	0.01279	0.00983	4.35646	4.36432	4.34004	**4.28970**	0.0000	0.0079	-0.0164	-0.0668
Boats	0.00000	0.00336	0.01167	0.00893	4.50582	4.42307	**4.40171**	4.40274	0.0000	-0.0827	-0.1041	-0.1031
Goldhill	0.00000	0.00000	0.00870	0.00895	5.13681	5.13323	**5.08080**	5.10719	0.0000	-0.0036	-0.0560	-0.0296
Average	0.00000	0.00400	0.02374	0.02245	4.98206	4.96107	4.94392	4.92686	0.0000	-0.0209	-0.0381	-0.0552

Table 3.4

A: Bit rate improvement in terms of the number of images when comparing Method-I with JPEG

	JPEG	Method-I	Total
No. of improved images out of 10	1	9	10

B: Bit rate improvement in terms of the number of images when comparing separately every sub method of Method-I with JPEG

	Run only	Run Length	Run in group (25:50:25)
No. of improved images out of 10	8	9	8

C : Bit rate improvement in terms of the number of images when comparing among all sub methods of Method-I.

	Run only	Run Length	Run in group (25:50:25)	Total
No. of improved images out of 10	2	3	5	10

The lowest bit rate achieved by the different methods for every image is shown in Figure 3.9. The performance of the various sub methods of Method-I over JPEG have been shown in Figure 3.10. The performance is measured by the difference in bit rate from JPEG. The negative difference indicates the reduction in bit rate. The zero line of Figure 3.10 is the JPEG line. The point above zero line means bit rate is reduced and below the zero line means bit rate is increased. From Table 3.3 and Figure 3.10, the following conclusion can be made for the first group of images when Method-I is used.

1. For peppers image bit rate does not improve by any of the sub methods of Method-I and can be seen from Figure 3.9 and 3.10. In all other images bit rate is improved by at least one of the sub methods of Method-I.
2. The maximum bit rate improvement is 0.15 bits/pixel for camera image and the second maximum bit rate improvement is 0.11 bits/pixel for couple image. Both improvement come from RO-3 sub method. The third maximum bit rate improvement which comes from RO-3 and RL sub methods is 0.10 bits/pixel for boats image.
3. The minimum improvement is .0010 bits/pixel by RO sub method for camera image which shows almost no improvement.
4. The average bit rate improvement is 0.0209 bits/pixel by RO, 0.0381 bits/pixel by RL and 0.0552 bits/pixel by RO-3.

5) The maximum average bit rate improvement comes from RO-3 and bit rates improves in the sequence RO→RL→RO-3.

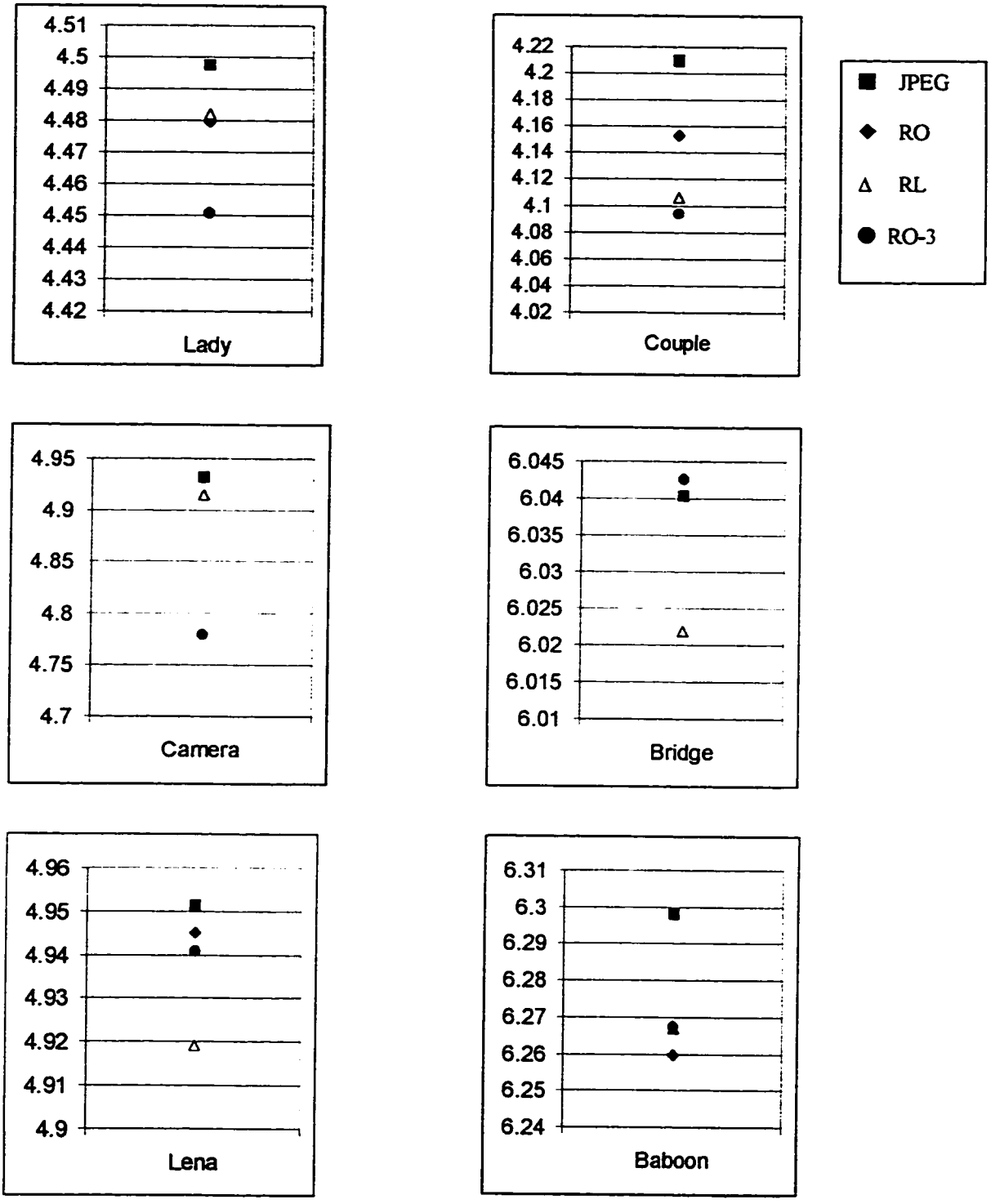


Figure 3.9: Lowest bit rate by different methods for different images.(continued)

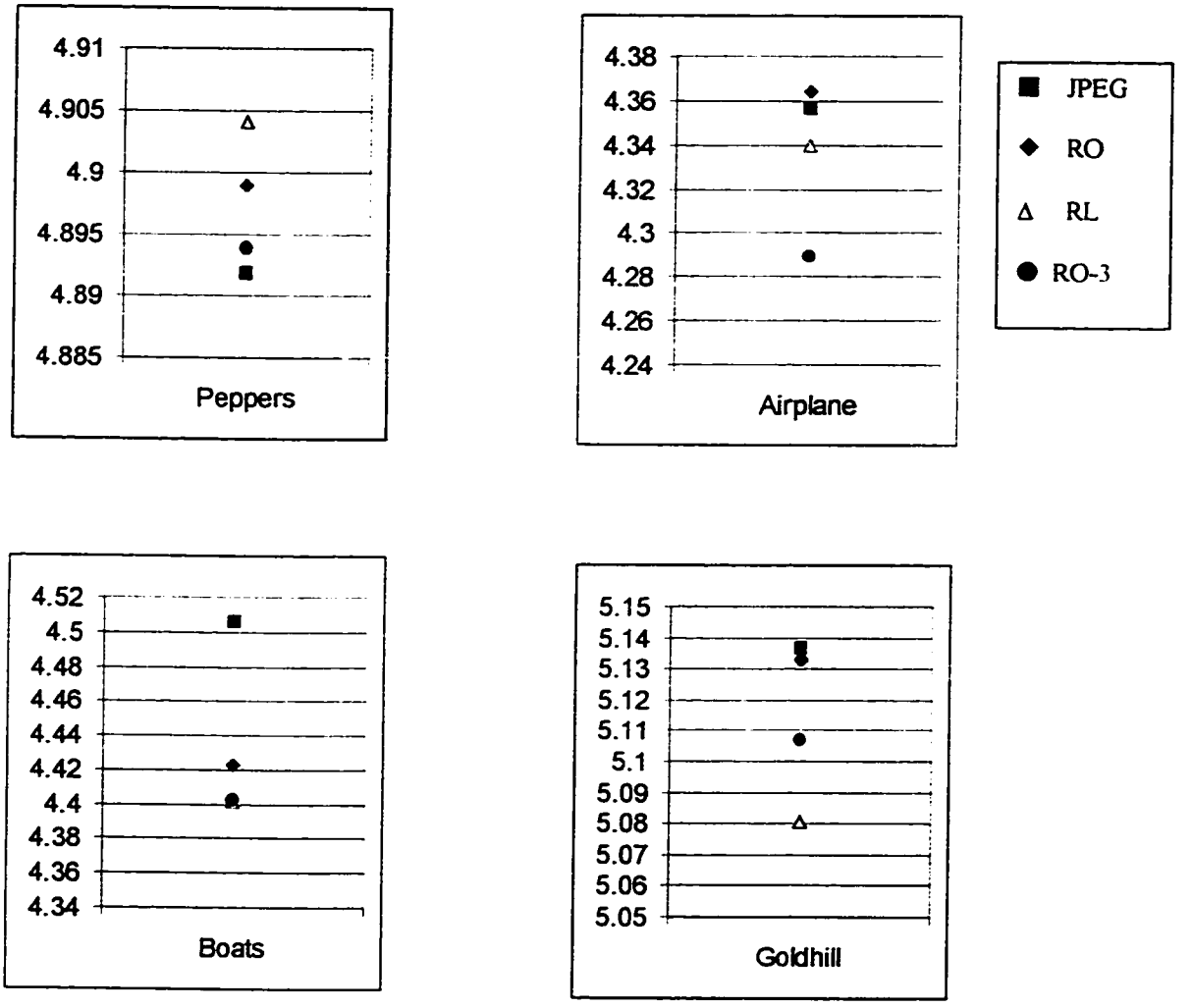


Figure 3.9: (Continued) Bit rate by different methods for different images

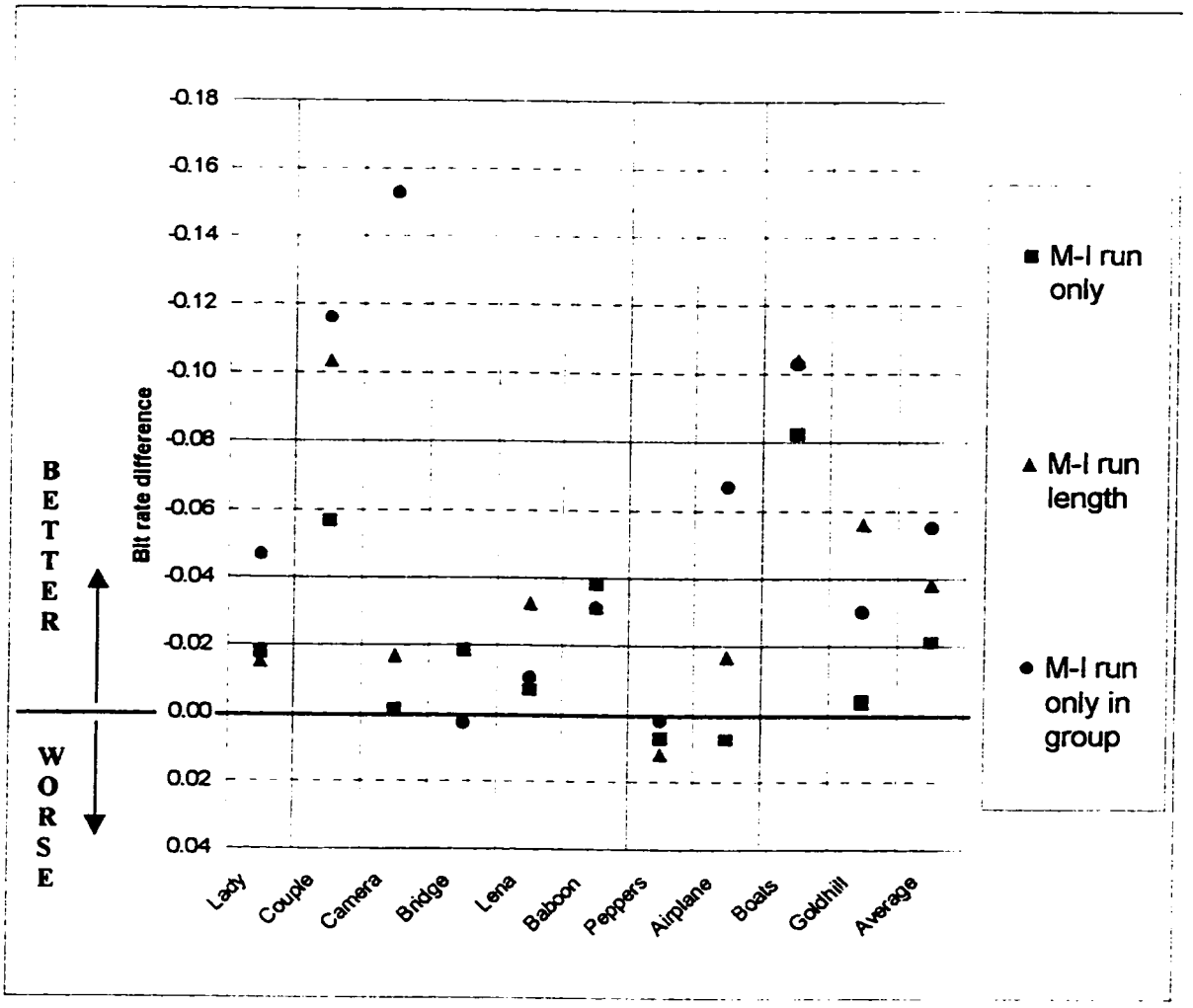


Figure 3.10: Difference in bit rate by various sub methods of Method-I from JPEG.

3.6.3 Effect of image size on overhead

The overhead of any image depends on the number of prediction errors present in the error image and the number of codes present in the code table. The image size has no influence on the number of prediction errors and little influence on the number of codes in the code table. The effect of the image size on the overhead is shown in Figure 3.11. The overhead per pixel is decreases when the image size is increases. When the image size is large, the overhead becomes negligibly small in comparison with the encoded bits/pixel and the difference in the overhead in different methods becomes smaller.

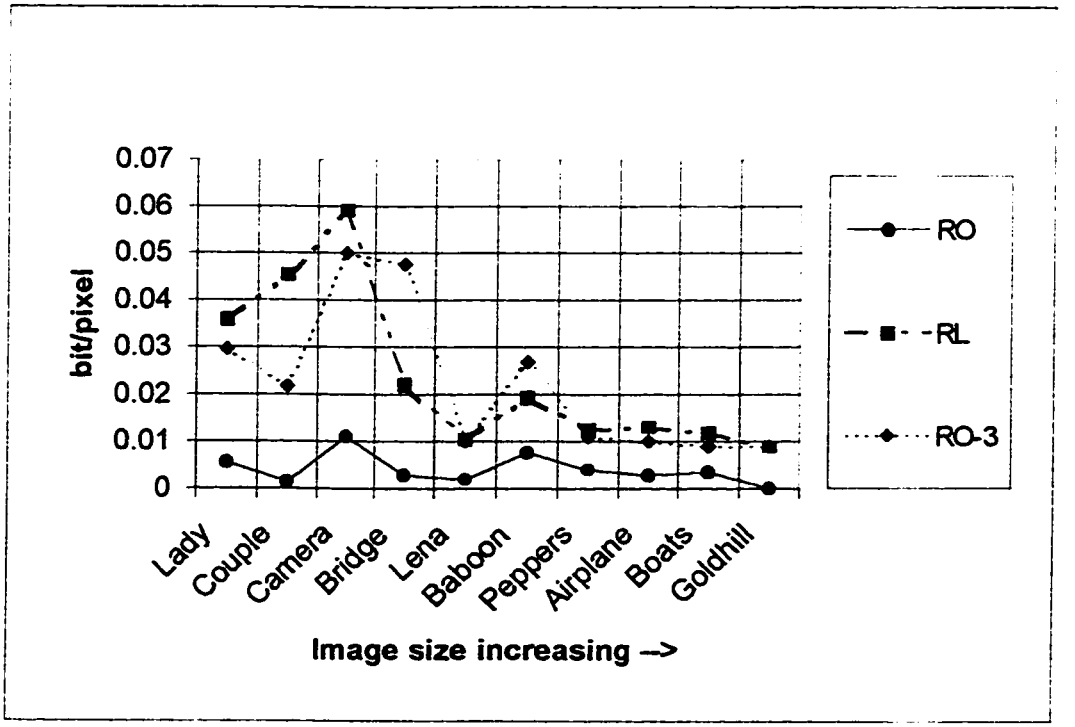


Figure 3.11: Overhead vs image size

3.6.4 Results of the second group of images

In this section, the results of the second group of images are presented. Usually lossless compression is required in medical images. These X-ray images are taken to verify the validity of Method-I on particular classes of image. These X-ray images are 236 pixel by 320 pixel and are smaller than the images of the first group. So, overhead will be higher for these small size images. Usually the size of medical images varies between 256 x 256 to 4094 x 4096 [3]. If Method-I works fine on these small size medical images, it is expected to work better for large size medical images because overhead per pixel is less for large size images.

Simulation results are presented for every X-ray image separately in the form of a table and a graph. There are two Tables A and B for every image. These results are generated using JPEG predictors. The second major column of the first table in every image shows the bit rate required to encode the image including the overhead. The results are presented for standard JPEG and Method-I. The results of Method-I are also presented in five different sub methods based on the different statistics.

RO, RL and RO-3 sub-methods, all are described in the previous section for the first group of images. Two additional sub methods are MG and MG-M. MG means that groups are chosen based on the nature of the zero-to-one ratio curve and the cumulative ones curve. The zero-to-one

ratio curve and the cumulative ones curve for four X-ray images are shown in Figure 3.12. The number of groups in MG is chosen between 3 to 5. The number of groups and the number of ones in the groups are chosen based on the zero-to-one ratio curve and the cumulative ones curve.

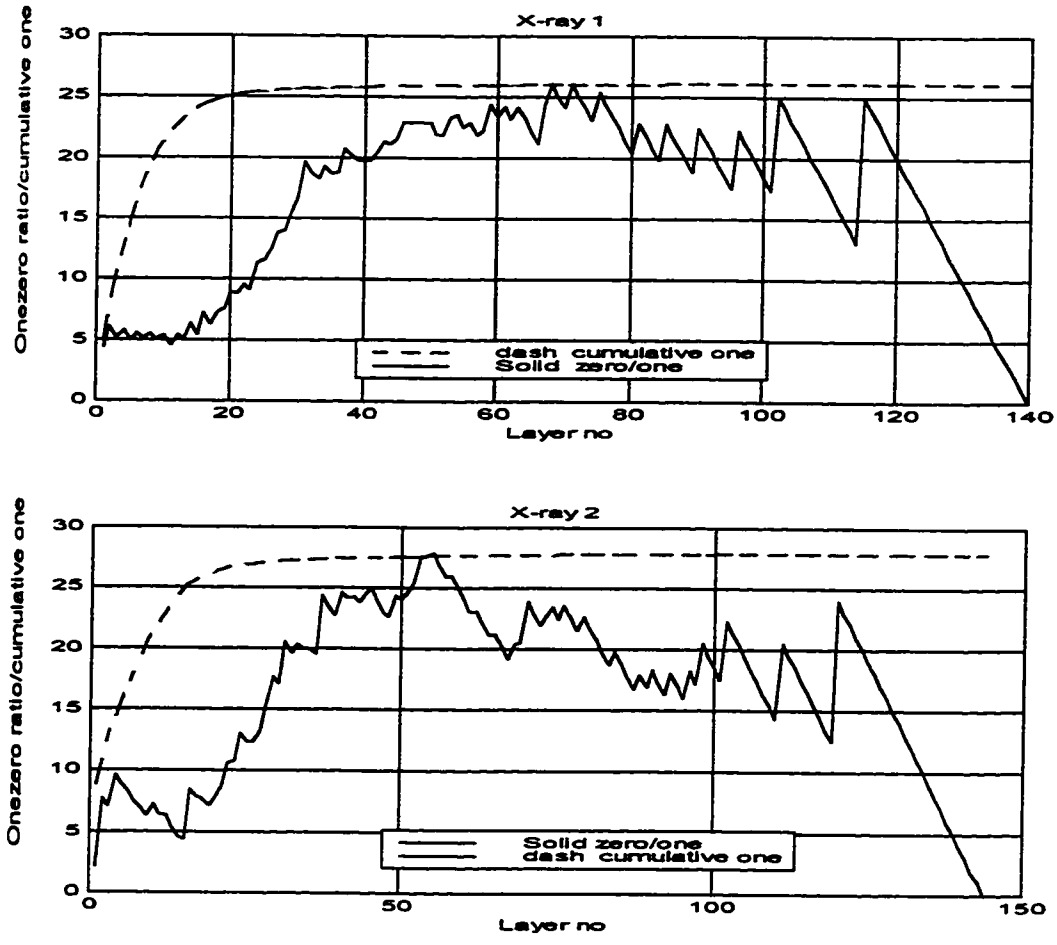


Figure 3.12: Plot of zero-to-one ratio curve and cumulative ones curve in the different layers of X-ray images

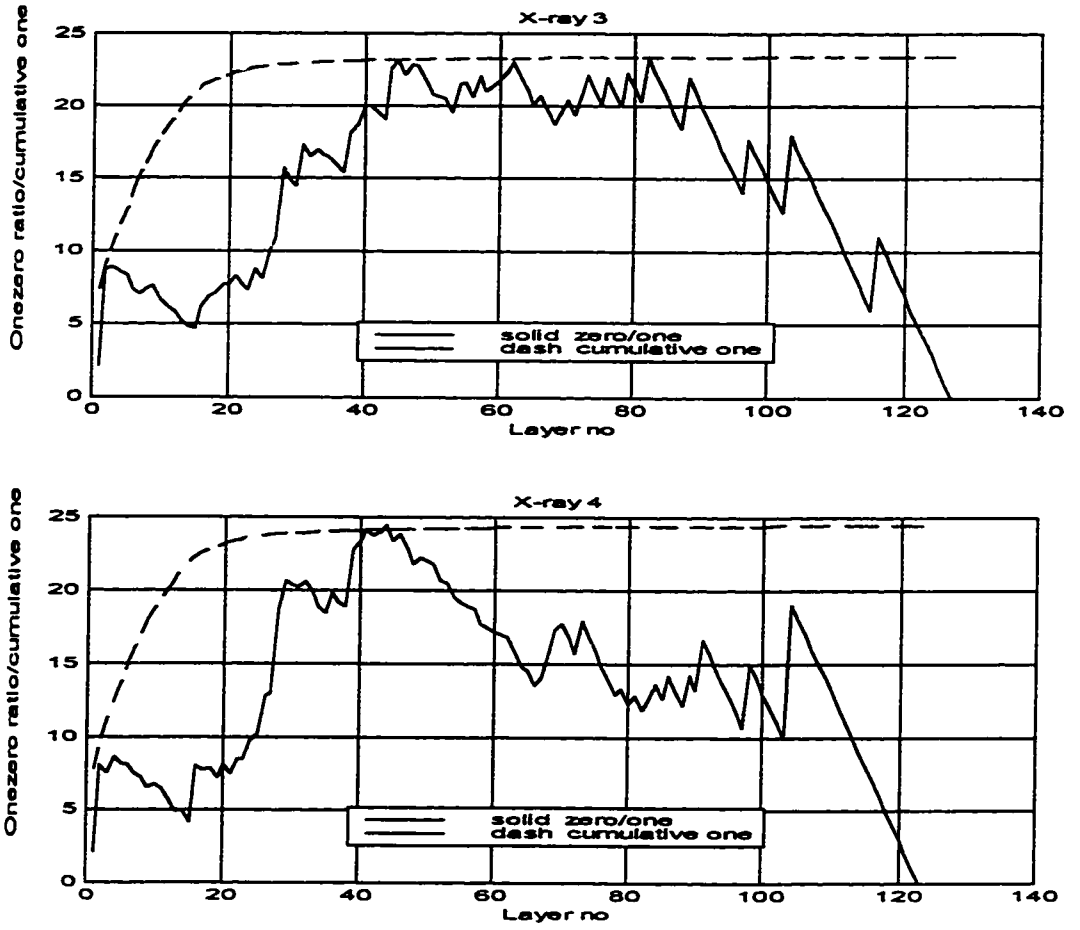


Figure 3.12: (cont) Plot of zero-to-one ratio curve and cumulative Ones curve in the different layers of X-ray images

In MG-M sub method, the number of groups and the number of 1's in the group are the same as the MG sub method. But the difference is that in MG-M, the first group is encoded by RL instead of RO. Based on the observation it is seen that zero-to-one ratio in the first group is always less than other groups. So high concentration of ones can be coded efficiently by RL rather than RO. The result obtained by MG-M is shown only for one predictor which produces minimum bit rate by MG.

3.6.4.1 X-ray 1

The simulation results of X-ray 1 are presented in Table 3.5. From the second major column of Table 3.5(A) it is seen that the bit rate improves from one sub method to another sub method. For the sub method MG, the bit rate improves for all predictors and MG-M shows further improvement. The following comments can be made for X-ray 1:

1. In Table 3.5(B), the compression ratio and the performance by different sub methods are shown. The performance is measured in terms of percentage change in compression ratio when compared with JPEG. Positive percentage change in compression ratio indicates improvement in compression ratio.
2. For the RO, RL and RO-3 sub methods, percentage change in compression ratio is positive for most of the predictors indicates improvement and negative for few predictors indicates no improvement or degradation. But sub method MG always shows an increase in compression ratio.
3. From Table 3.5(B) the maximum improvement in compression ratio is 2.36%. The percentage change in compression ratio when compared to JPEG is plotted in Figure 3.13. From this figure it is seen that curve for MG sub method lies in top and the point indicating by MG-M shows the highest improvement.

Table 3.5: Simulation results for X-ray 1 image

(A)

X-ray 1	Entropy (bits/pixel)						Bit rate with overhead (bits/pixel)					
	JPEG	Method-I					JPEG	Method-I				
Predictor	Lossless JPEG	Run only (RO)	Run leng (RL)	Run in grp (RO-3) 25:50:25	MG	MG-M	Lossless JPEG	Run only (RO)	Run leng (RL)	Run in grp (RO-3) 25:50:25	MG	MG-M
A	4.09515	4.18579	4.15003	4.08235	4.05446		4.12521	4.21052	4.2006	4.13000	4.10221	
B	4.03940	3.99665	3.97887	3.92656	3.91317	3.90883	4.06247	4.02004	4.0285	3.97926	**3.98003**	3.96859
C	4.38479	4.34850	4.32867	4.26990	4.25605		4.40128	4.37708	4.3870	4.38007	4.38089	
A+B-C	4.27654	4.29664	4.28496	4.27805	4.26205		4.31946	4.32963	4.3284	4.31876	4.30205	
A+(B-C)/2	4.12426	4.14000	4.12858	4.12618	4.11875		4.16726	4.17886	4.1709	4.16563	4.15537	
B+(A-C)/2	4.10640	4.06320	4.04984	4.04182	4.03388		4.15112	4.10098	4.0967	4.08839	4.08576	
(A+B)/2	4.01258	3.96061	3.94994	3.95142	3.94235		**4.05100**	3.98876	3.9977	3.99774	3.98924	

(B)

X-ray 1	Compression ratio						Performance in terms of change in compression ratio over JPEG (%)					
	JPEG	Method- I					JPEG	Method I				
Predictor	Lossless JPEG	Run only (RO)	Run leng (RL)	Run in grp (RO-3) 25:50:25	MG	MG-M	Lossless JPEG	Run only (RO)	Run leng (RL)	Run in grp (RO-3) 25:50:25	MG	MG-M
A	1.93930	1.90000	1.90448	1.93705	1.95017		0.00000	-2.0261	-1.79518	-0.11594	0.56066	
B	1.96925	1.99003	1.98582	2.01042	2.01004	2.01583	0.00000	1.05538	0.84154	2.09114	2.07145	2.36556
C	1.81765	1.82770	1.82355	1.82646	1.82611		0.00000	0.55290	0.32429	0.48428	0.46551	
A+B-C	1.85208	1.84773	1.84825	1.85238	1.85958		0.00000	-0.2349	-0.20691	0.01623	0.40473	
A+(B-C)/2	1.91973	1.91440	1.91805	1.92048	1.92522		0.00000	-0.2776	-0.08749	0.03909	0.28609	
B+(A-C)/2	1.92719	1.95075	1.95277	1.95676	1.95802		0.00000	1.22268	1.32721	1.53442	1.59967	
(A+B)/2	1.97482	2.00563	2.00111	2.00113	2.00540		0.00000	1.56034	1.33101	1.33224	1.54819	

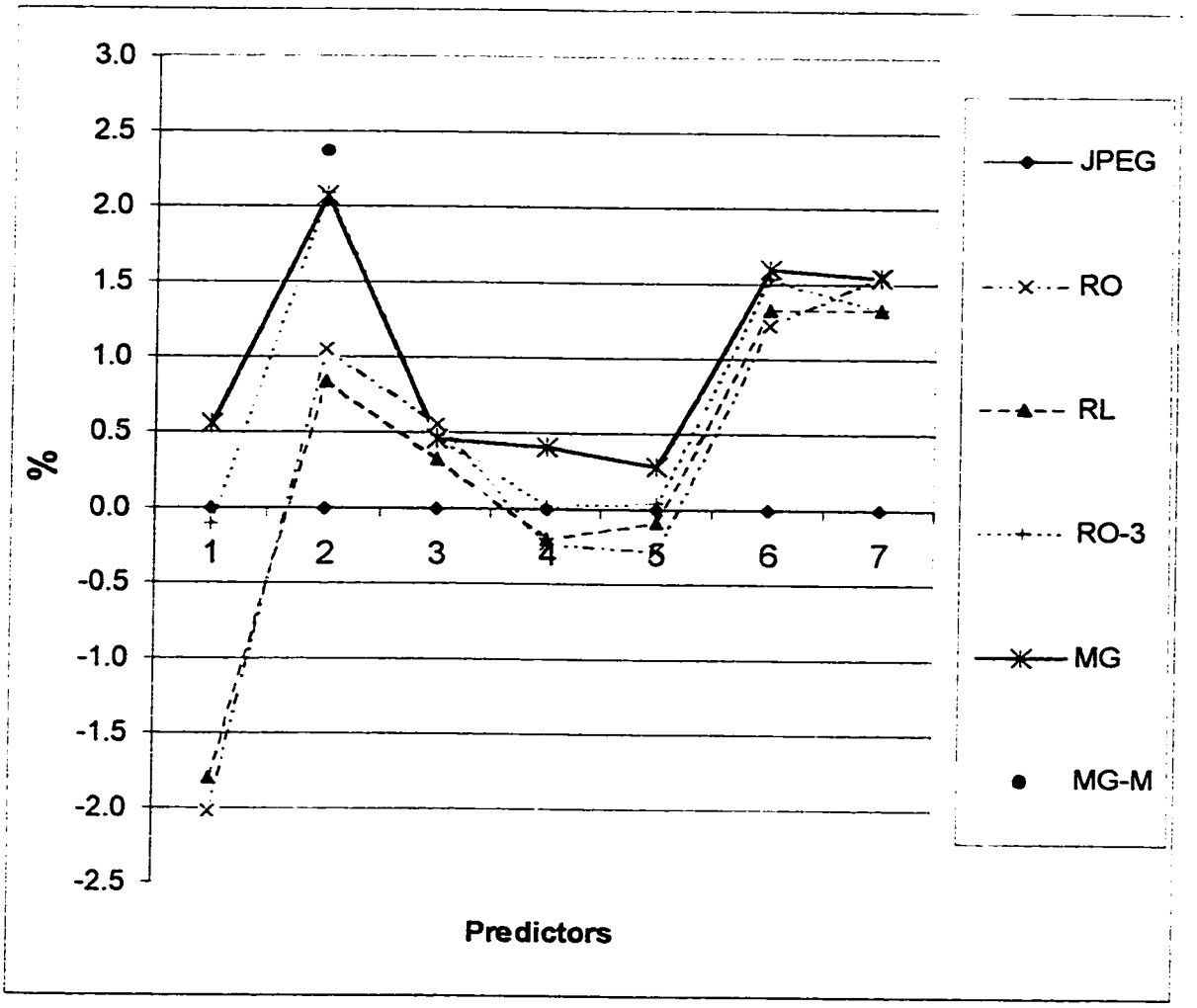


Figure 3.13: Percentage change in compression ratio in X-ray 1 by different methods for different predictors taken JPEG as a reference

3.6.4.2 X-ray 2

Simulation results of X-ray 2 are presented in Table 3.6. The second major column of Table 3.6(A) shows that the bit rate improves for all predictors by MG. MG-M shows further improvement in bit rate for the predictor which gives the lowest bit rate by MG. The following comments can be made for X-ray 2:

1. The performance is presented in Table 3.6(B). Table 3.6(B) shows the improvement in compression ratio from one sub method to another sub method.
2. The maximum improvement in compression ratio is 3.54% by MG and 3.58% by MG-M.
3. The maximum compression ratio by JPEG is achieved for predictor 1 or A but the maximum compression ratio by MG-M is achieved for predictor B. Therefore results show that predictor which is suitable for one method may not be suitable for another method.
4. The change in compression ratio by different sub methods of Method-I when compared to JPEG is plotted Figure 3.14. Five curves show the five different sub methods and JPEG lies in the zero line. Curves go up from one sub method to another sub method.
5. The curve for sub method MG lies on top. MG-M shows by single dot for predictor B and it is the highest improvement. So encoding by different groups can improve compression ratio.

Table 3.6: Simulation results for X-ray 2 image

(A)

X-ray 2	Entropy (bits/pixel)						Bit rate with overhead (bits/pixel)					
	JPEG	Method-I					JPEG	Method-I				
Predictor	Lossless JPEG	Run only (RO)	Run leng (RL)	Run in grp (RO-3) 25:50:25	MG	MG-M	Lossless JPEG	Run only (RO)	Run leng (RL)	Run in grp (RO-3) 25:50:25	MG	MG-M
A	3.95218	4.04282	4.00951	3.95214	3.91112		**3.99420**	4.06356	4.0569	4.00418	3.96842	
B	3.98568	3.90628	3.88961	3.84032	3.82287	3.82002	4.01840	3.92937	3.9377	3.89062	**3.88086**	3.87936
C	4.36541	4.28648	4.26626	4.21255	4.19439		4.39103	4.31409	4.3252	4.27164	4.25067	
A+B-C	4.19425	4.21539	4.20573	4.19646	4.18247		4.23311	4.25254	4.2486	4.23698	4.22116	
A+(B-C)/2	4.04182	4.05943	4.05253	4.05147	4.03619		4.08953	4.09675	4.0934	4.09241	4.08172	
B+(A-C)/2	4.08399	4.01845	4.00668	3.99836	3.98670		4.12971	4.05492	4.0541	4.04432	4.03775	
(A+B)/2	4.00123	3.92298	3.91247	3.91249	3.89609		4.03988	3.95062	3.9599	3.95522	3.94008	

(B)

X-ray 2	Compression ratio						Performance in terms of change in compression ratio over JPEG (%)					
	JPEG	Method-I					JPEG	Method-I				
Predictor	Lossless JPEG	Run only (RO)	Run leng (RL)	Run in grp (RO-3) 25:50:25	MG	MG-M	Lossless JPEG	Run only (RO)	Run leng (RL)	Run in grp (RO-3) 25:50:25	MG	MG-M
A	2.00290	1.96872	1.97195	1.99791	2.01592		0.00000	-1.7067	-1.54568	-0.24917	0.64973	
B	1.99084	2.03595	2.03164	2.05623	2.06140	2.06219	0.00000	2.26578	2.04951	3.28441	3.54406	3.58397
C	1.82190	1.85439	1.84959	1.87282	1.88205		0.00000	1.78349	1.52006	2.79491	3.30195	
A+B-C	1.88986	1.88123	1.88296	1.88814	1.89521		0.00000	-0.4569	-0.36550	-0.09123	0.28314	
A+(B-C)/2	1.95622	1.95277	1.95433	1.95484	1.95996		0.00000	-0.1763	-0.09661	-0.07044	0.19125	
B+(A-C)/2	1.93718	1.97291	1.97331	1.97808	1.98130		0.00000	1.84451	1.86475	2.11136	2.27751	
(A+B)/2	1.98026	2.02500	2.02020	2.02264	2.03042		0.00000	2.25932	2.01730	2.14035	2.53303	

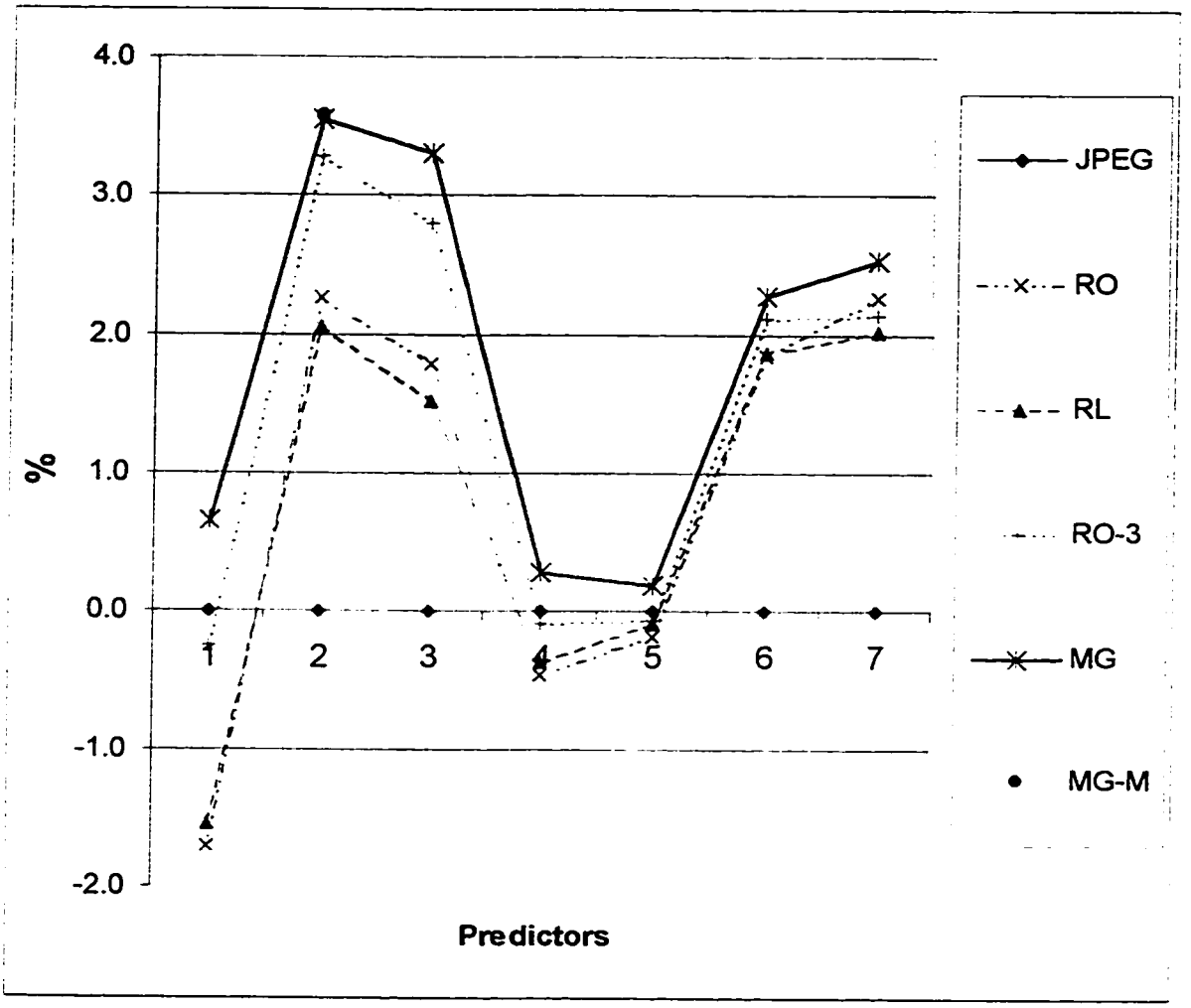


Figure 3.14: Percentage change in compression ratio in X-ray 2 by different methods for different predictors taken JPEG as a reference

3.6.4.3 X-ray 3

Simulation results of X-ray 3 are presented in Table 3.7. The second major column of Table 3.7(A) shows that the bit rate improves for all predictors by MG sub method and MG-M sub method shows further improvement in bit rate for the predictor which gives the lowest bit rate by MG sub method. The following observation can be made for X-ray3:

1. The performance is presented in Table 3.7(B). The second major column of Table 3.7(B) shows the improvement in compression ratio from one sub method to another sub method.
2. The maximum improvement in compression ratio is 3.357% by MG and 3.44% by MG-M.
3. The maximum compression ratio by JPEG is achieved for predictor 1 or A, but the maximum compression ratio achieved by MG is for predictor 7 or (A+B)/2. So results show that predictor which is good for one method may not be good for the other.
4. The percentage change in compression ratio is plotted in Figure 3.15. Five curves show five different sub methods of Method-I and JPEG lies in the zero line. Curves go up from one sub method to another. The sub method MG-M shows by single dot for predictor (A+B)/2 and it is the highest improvement. Therefore encoding the bit stream by different groups can improve compression ratio.

Table 3.7: Simulation results for X-ray 3 image

(A)

X-ray 3	Entropy (bits/pixel)						Bit rate with overhead (bits/pixel)					
	JPEG	Method I					JPEG	Method I				
Predictor	Lossless JPEG	Run only (RO)	Run leng {RL)	Run in grp (RO-3) 25:50:25	MG	MG-M	Lossless JPEG	Run only (RO)	Run leng {RL)	Run in grp (RO-3) 25:50:25	MG	MG-M
A	3.94241	4.02067	3.97597	3.93581	3.89767		**3.97394**	4.04100	4.0243	3.98845	3.95361	
B	4.04853	3.95901	3.92982	3.89024	3.86998		4.06930	3.97619	3.9838	3.94460	3.93710	
C	4.34535	4.25207	4.22721	4.12428	4.17622		4.36154	4.28026	4.2887	4.24216	4.23585	
A+B-C	4.18863	4.21036	4.19750	4.18925	4.17487		4.22081	4.24927	4.2407	4.22836	4.21484	
A+(B-C)/2	4.01363	4.02564	4.01369	4.01377	4.00386		4.05634	4.06291	4.0551	4.05665	4.04924	
B+(A-C)/2	4.08880	4.01797	4.00243	3.99607	3.98103		4.13294	4.05429	4.0518	4.04570	4.03853	
(A+B)/2	3.96813	3.87901	3.86520	3.87128	3.86139	3.85580	4.00785	3.90423	**3.8670**	3.91114	3.90374	**3.91185**

(B)

X-ray 3	Compression ratio						Performance in terms of change in compression ratio over JPEG (%)					
	JPEG	Method I					JPEG	Method I				
Predictor	Lossless JPEG	Run only (RO)	Run leng {RL)	Run in grp (RO-3) 25:50:25	MG	MG-M	Lossless JPEG	Run only (RO)	Run leng {RL)	Run in grp (RO-3) 25:50:25	MG	MG-M
A	2.01312	1.97971	1.98791	2.00579	2.02347		0.0000	-1.6594	-1.2520	-0.3639	0.5143	
B	1.96594	2.01198	2.00810	2.02809	2.03195		0.0000	2.3418	2.1447	3.1612	3.3579	
C	1.83421	1.86905	1.86535	1.88583	1.88864		0.0000	1.8990	1.6977	2.8143	2.9673	
A+B-C	1.89537	1.88268	1.88647	1.89199	1.89806		0.0000	-0.6697	-0.4698	-0.1786	0.1416	
A+(B-C)/2	1.97222	1.96903	1.97279	1.97207	1.97568		0.0000	-0.1617	0.0288	-0.0077	0.1753	
B+(A-C)/2	1.93567	1.97322	1.97443	1.97741	1.98092		0.0000	1.9399	2.0024	2.1564	2.3377	
(A+B)/2	1.99608	2.04906	2.06875	2.04544	2.04931	2.04507	0.0000	2.6539	3.6407	2.4728	2.6668	2.4804

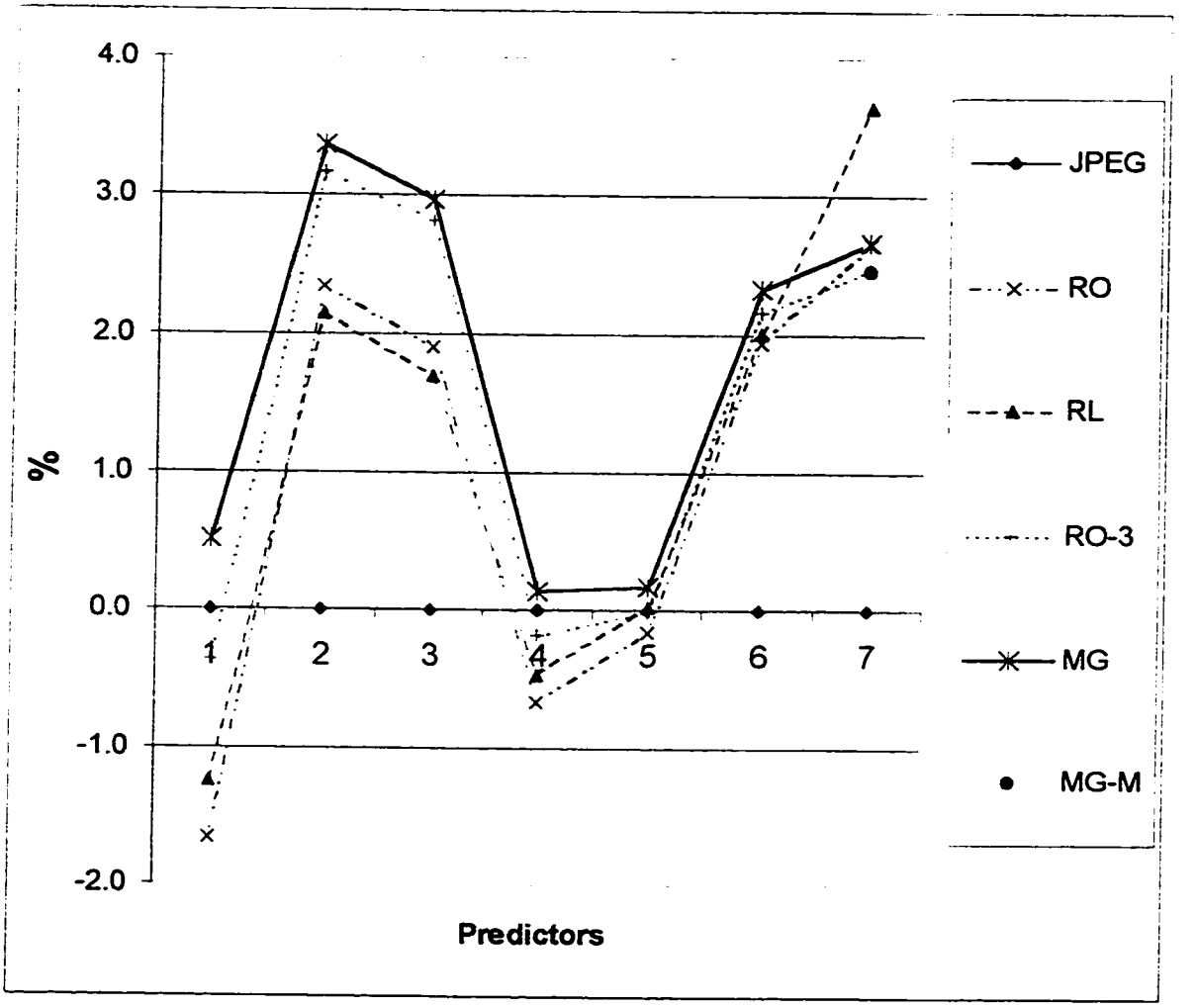


Figure 3.15: Percentage change in compression ratio in X-ray 3 by different methods for different predictors taken JPEG as a reference

3.6.4.4 X-ray 4

Simulation results of X-ray 4 are presented in Table 3.8. The second major column of Table 3.8(A) shows that the bit rate improves for all predictors by MG and MG-M shows further improvement in bit rate for the predictor which produces the lowest bit rate by MG. The following observation can be made from Table 3.8(B) and Figure 3.16 for X-ray 4:

1. The second major column of Table 3.8(B) shows that the compression ratio improves from one sub method to another sub method.
2. The maximum improvement in compression ratio is 3.64% by MG and 3.73% by MG-M.
3. The maximum compression ratio by JPEG is achieved for predictor 1 or A but the maximum compression ratio achieve by MG for predictor 2 or B. So the results show that one predictor does not show always best result for all methods.
4. The percentage change in compression ratio by different sub methods of Method-I is plotted in Figure 3.16. Five curves for five different sub methods of method-I. The curve for MG lies on the top. The sub method MG-M shows by single dot for predictor B indicates the highest improvement. So encoding bit stream by making suitable different groups can improve compression ratio.

Table 3.8: Simulation results for X-ray 4 image

(A)

X-ray 4	Entropy (bits/pixel)						Bit rate with overhead (bits/pixel)					
	JPEG	Method-I					JPEG	Method-I				
Predictor	Lossless JPEG	Run only (RO)	Run leng (RL)	Run in grp (RO-3) 25:50:25	MG	MG-M	Lossless JPEG	Run only (RO)	Run leng (RL)	Run in grp (RO-3) 25:50:25	MG	MG-M
A	3.92303	3.99410	3.94886	3.91819	3.87253		**3.96169**	4.01240	3.9945	3.96247	3.92299	
B	3.96036	3.87900	3.85901	3.81979	3.79106	3.80991	3.99241	3.90533	3.9086	3.86538	**3.85205**	3.84858
C	4.32276	4.23364	4.20969	4.16396	4.14402		4.34637	4.26068	4.2700	4.21997	4.20927	
A+B-C	4.16366	4.18068	4.16594	4.16005	4.14603		4.19606	4.22241	4.2115	4.20064	4.18888	
A+(B-C)/2	3.99375	4.00436	3.98819	3.99328	3.98329		4.03444	4.04085	4.0374	4.03642	4.02777	
B+(A-C)/2	4.04356	3.97387	3.95814	3.95145	3.94108		4.08960	4.00635	4.0088	3.99929	3.99799	
(A+B)/2	3.95029	3.86109	3.84412	3.85211	3.84051		3.98930	3.88696	3.8921	3.89217	3.87770	

(B)

X-ray 4	Compression ratio						Performance in terms of change in compression ratio over JPEG (%)					
	JPEG	Method I					JPEG	Method I				
Predictor	Lossless JPEG	Run only (RO)	Run leng (RL)	Run in grp (RO-3) 25:50:25	MG	MG-M	Lossless JPEG	Run only (RO)	Run leng (RL)	Run in grp (RO-3) 25:50:25	MG	MG-M
A	2.01934	1.99382	2.00271	2.01894	2.03926		0.0000	-1.2638	-0.8233	-0.0198	0.9866	
B	2.00380	2.04848	2.04675	2.06965	2.07682	2.07869	0.0000	2.2299	2.1435	3.2864	3.6439	3.7371
C	1.84062	1.87764	1.87351	1.89575	1.90057		0.0000	2.0113	1.7870	2.9952	3.2572	
A+B-C	1.90655	1.89465	1.89953	1.90447	1.90982		0.0000	-0.6239	-0.3680	-0.1091	0.1714	
A+(B-C)/2	1.98293	1.97978	1.98147	1.98195	1.98621		0.0000	-0.1586	-0.0733	-0.0491	0.1655	
B+(A-C)/2	1.95618	1.99683	1.99558	2.00035	2.00100		0.0000	2.0780	2.0140	2.2581	2.2913	
(A+B)/2	2.00536	2.05816	2.05544	2.05541	2.06308		0.0000	2.6329	2.4972	2.4955	2.8780	

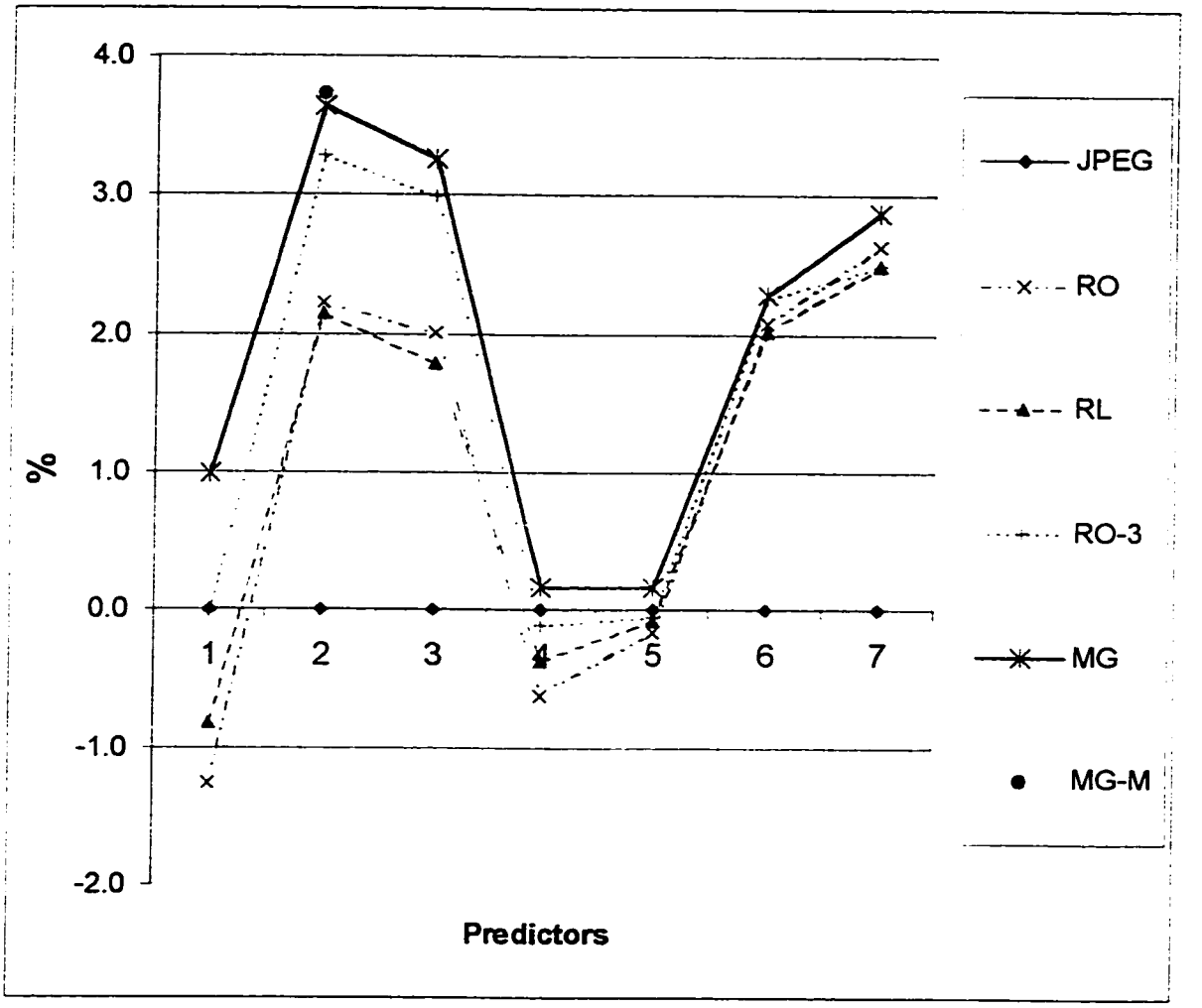


Figure 3.16: Percentage change in compression ratio in X-ray 4 by different methods for different predictors taken JPEG as a reference

3.6.5 Summary of the results for X-ray images

A summary of the results for the second group of images are presented in Table 3.9. This table shows the bit rate achieved by different methods and the improvement in compression ratio when compared with JPEG. The bit rate here is achieved by the best predictor and is calculated including the overhead. The best predictor is the predictor that gives the minimum bit rate out of all seven predictors. Table 3.9 presents the minimum bit rate which can be achieved by different methods for every image. Results show that bit rate improves by every sub methods of Method-I. The following conclusions can be made for the X-ray images:

1. Compression ratio improves for all of the sub methods of Method-I when compared to JPEG.
2. Three images out of four show the highest improvement in compression ratio by MG-M and one shows the highest improvement by RL. These highest improvements are bold faced in the second major column of Table 3.9.
3. In average MG-M shows the best performance by 2.38% followed by MG which shows 2.33%. The minimum improvement is 1.33% and the maximum is 2.96%.

Table 3.9: Summary of the results for X-ray images

Image 236x320	Bit rate with overhead (bits/pixel)						Improvement in compression ratio (%)					
	JPEG	Method I					JPEG	Method I				
	Lossless JPEG	Run only (RO)	Run leng (RL)	Run in grp (RO-3) 25:50:25	MG	MG-M	Lossless JPEG	Run only (RO)	Run leng (RL)	Run in grp (RO-3) 25:50:25	MG	MG-M
X-ray 1	**4.05100**	3.9887	3.99779	3.97926	3.98003	**3.96859**	0	1.56191	1.3309	1.80285	1.78315	**2.07656**
X-ray 2	**3.99420**	3.92937	3.93770	3.89062	3.88086	**3.87936**	0	1.64988	1.4348	2.66230	2.92049	**2.96028**
X-ray 3	**3.97394**	3.90423	**3.86706**	3.91114	3.90374	3.91185	0	1.78550	**2.7638**	1.60567	1.79828	1.58723
X-ray 4	**3.96169**	3.88696	3.89211	3.86538	3.85205	**3.84858**	0	1.92258	1.7877	2.49160	2.84628	**2.93901**
Average	**3.99520**	3.92730	3.92370	3.91160	3.90420	**3.90210**	0	1.72873	1.8233	2.13742	2.33180	**2.38622**

4. The sequence of improvement among the different sub methods of Method-I is RO→ RL→ RO-3 → MG → MG-M and these sequence can be observed from the last row of Table 3.9.

5. From the second major column of Table 3.9 it can be seen that improvement in compression ratio by Method-I varies from 1% to 3%.

6. Choosing any sub method will improve the performance but MG-M can produce the best performance.

Percentage improvement in compression ratio by different sub methods of Method-I are presented as a bar graph in Figure 3.17. In

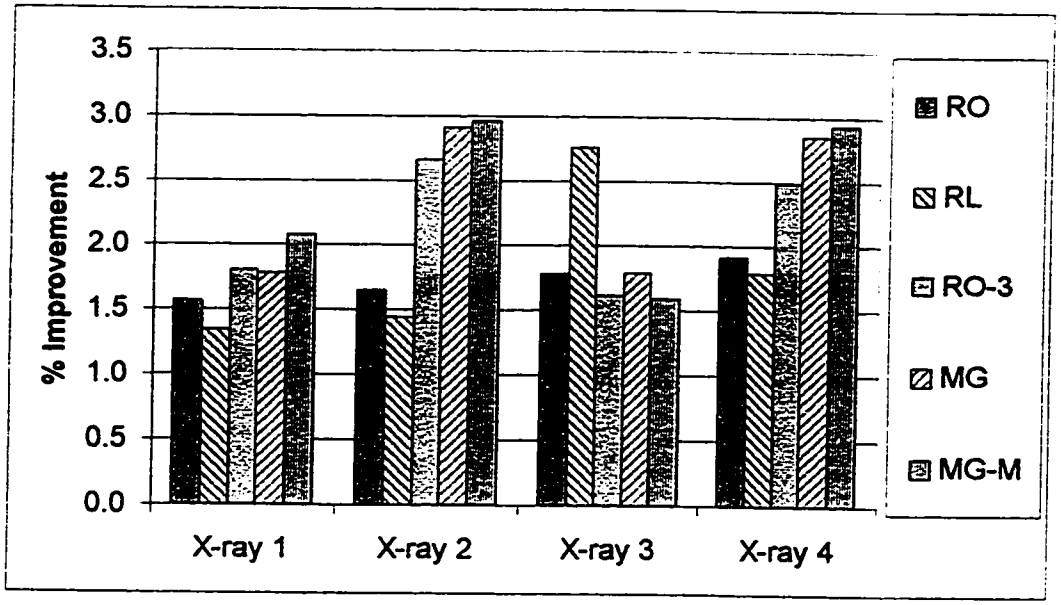


Figure 3.17: Improvement in compression ratio by different sub methods of Method-I for X-ray images

most of the sub methods, the bar for MG-M is greater than other methods, it means that among the different sub methods, MG-M performs better than any other sub method.

The lowest bit rate obtained by different methods for X-ray images is shown in Figure 3.18. This figure shows the clear cut difference in bit

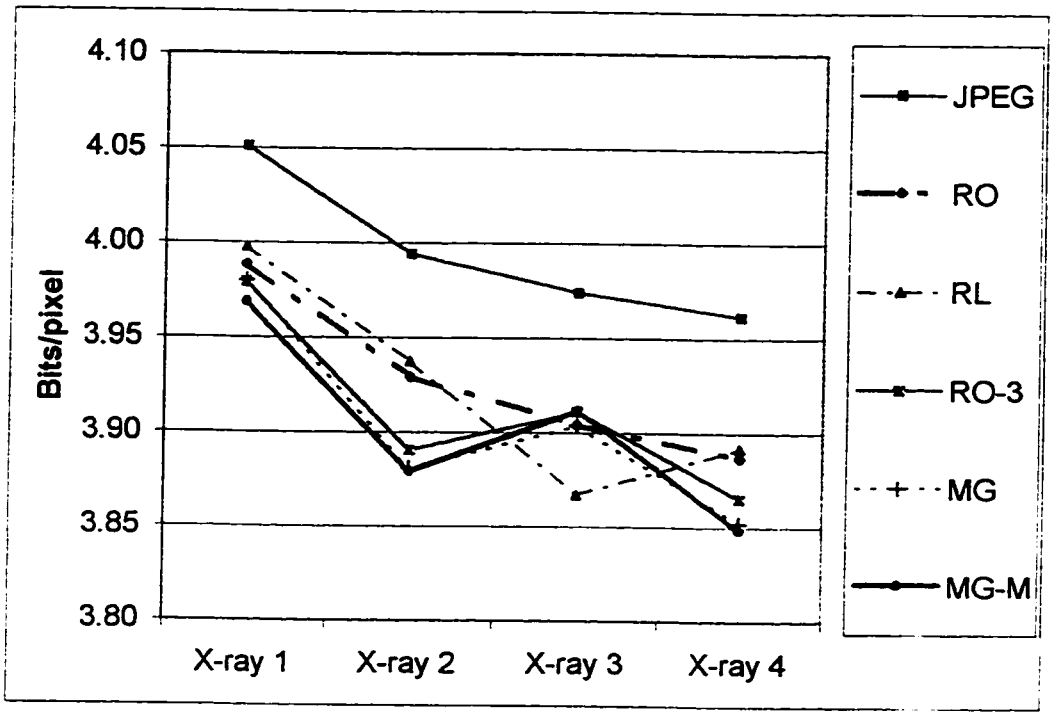


Figure 3.18: Performance of Method-I over JPEG in terms of bit rate for X-ray images.

rate between JPEG and Method-I. All the curves for the different sub-methods of Method-I lie below the JPEG line, it indicates that bit rate is reduced by Method-I.

Lossless image compression is almost saturated in terms of compression ratio and this ratio varies within a few percentage of order of two. Even though improvement by Method-I is small but it reduces the bit rate from an unbreakable barrier which is the entropy of the error image.

Usually research in literature is focused on finding the best prediction method to decorrelate the prediction errors which will lower the entropy. Incorporating Method-I to model the prediction errors will reduce the bit rate further irrespective of the prediction method used.

3.6.6 Complexity

Method- I reduces bit rate at the cost of complexity. Complexity does not always mean improvement over bit rate. Complexity can be accepted in some cases to increase compression ratio because of the smaller compression ratio for lossless compression. Method-I shows two types of complexity. These are time and memory. Time for encoding and decoding has increased because of the extra time needed for modeling prediction errors. The memory complexity arises because the prediction

errors have to be stored before modeling and bit streams also have to be stored before encoding the bit stream.

Table 3.10 shows the time required by JPEG and Method-I. Time for compression by JPEG consists of three elements and these are (i) time for generating prediction errors (ii) Time for generating Huffman code table based on the statistics (iii) Time for encoding the prediction errors using the code table generated in the previous step.

Time required for compression by Method-I consists of three main components and these are (I) Time for making bit-stream from prediction errors (ii) Time for taking statistics from bit stream and generating Huffman code table (iii) Time for encoding the bit stream based on the code table generated in the previous step.

Time for decompression by JPEG is the time for decoding the compressed bit stream to generate prediction errors and generating original pixel values from these prediction errors. Time for decompression by Method-I consists of time required to generate bit stream of many layers from the compressed bit stream and generating prediction errors from the bit stream of many layers and recovering the original pixel values from the prediction errors. Generating the Huffman code table from the encoded bit stream is also considered here.

Table 3.10 shows that compression and decompression time of group-1 images by Method-I increased compared to JPEG. Table 3.11

Table 3.10: Encoding and Decoding time required by JPEG and Method-I for Group 1 images

	JPEG					METHOD-I				
	Compression				Decompression	Compression				Decompression
IMAGE	Time for Prediction Sec.	Time for Making Code table Sec.	Encoding Time Sec.	Total Compression time Sec.	Decoding Time Sec.	Time for making bit stream Sec.	Time for taking statistics Sec.	Time for encoding bit stream Sec.	Total compression time Sec.	Decoding Time Sec.
Lady	0.17	0.99	0.22	1.38	11.59	7.08	0.27	14.22	21.57	41.47
Couple	0.11	1.27	0.16	1.54	10.77	6.92	0.22	12.58	19.72	33.78
Camera	0.11	2.25	0.22	2.58	13.46	12.52	0.39	12.41	25.32	52.51
Bridge	0.11	1.15	0.28	1.54	17.08	10.11	0.60	16.26	26.97	54.65
Lena	0.50	3.85	0.77	5.12	54.05	34.44	1.16	10.93	46.53	207.73
Baboon	0.49	5.33	1.27	7.09	74.75	44.27	2.80	22.35	69.42	222.61
Peppers	0.55	4.95	0.77	6.27	54.10	34.27	1.15	11.70	47.12	160.99
Airplane	0.55	4.56	0.66	5.77	46.08	29.05	0.88	10.77	40.70	133.91
Boats	2.20	8.24	1.15	11.59	75.75	52.18	1.42	17.41	71.01	218.80
Goldhill	1.65	10.54	1.48	13.67	90.19	93.86	2.03	12.74	108.63	266.72

Table 3.11: Encoding and Decoding time required by JPEG and Method-I for group-2 images

	JPEG					METHOD-I				
	Compression				Decompression	Compression				Decompression
IMAGE	Time for Prediction Sec.	Time for Making Code table Sec.	Encoding Time Sec.	Total Compression time Sec.	Decoding Time Sec.	Time for making bit stream Sec.	Time for taking statistics Sec.	Time for encoding bit stream Sec.	Total compression time Sec.	Decoding Time Sec.
X-ray1	0.11	1.04	0.16	1.31	11.92	7.20	0.16	15.21	22.57	39.16
X-ray2	0.16	1.42	0.17	1.75	11.60	7.36	0.22	13.84	21.42	39.33
X-ray3	0.16	0.77	0.16	1.09	11.37	6.48	0.22	11.81	18.51	39.38
X-ray4	0.17	1.10	0.17	1.44	11.70	10.99	0.22	12.35	23.56	41.09

Table 3.12: Memory requirement by Method-I

IMAGES Of Group 1	Number of bits required to store bit stream	Extra memory required to store bit stream in terms of memory required by original image (Times)	IMAGES Of Group 2	Number of bits required to stored bit stream	Extra memory required to store bit-stream in terms of memory required by original image (Times)
Lady	613146	1.17	X-ray1	510304	0.84
Couple	521207	0.99	X-ray2	565304	0.94
Camera	1055504	2.01	X-ray3	540455	0.89
Bridge	1597538	3.05	X-ray4	538702	0.89
Lena	3141099	1.50			
Baboon	7648448	3.65			
Peppers	3055252	1.46			
Airplane	2316670	1.10			
Boats	3767497	1.14			
Goldhill	6571887	1.98			

shows the time required for group-2 images. Simulations are done using visual C++ and 133 MHz Pentium with 32MB RAM. Programs of Method-I is not optimized for reducing the time required to run different programs. So, the actual time would be less than the time presented in the table. Table 3.12 shows the extra memory required by Method-I to store the bit stream of many layers before encoding. Memories are calculated in terms of memory required to store the original image. Table 3.12 shows that extra memory varies from .99 to 3.65 times of memory required by original images. Baboon image is a high detail image and it takes the highest memory. But group 2 images show that maximum extra memory required is the same as the memory required by the original image.

Table 3.10 and 3.11 show that Method-I introduces extra time for encoding and decoding and Table 3.12 show that the amount of extra memory required to store the bit stream of many layers before encoding. So Method-I is good for off-line use and higher-speed machine with larger RAM size can reduce these complexities.

3.6.7 Observation of features and its impact on results

01: Goldhill, Peppers, Baboon, Bridge, Lena images are high detail images and the bit rate reduction by Method-I for these images are less compared to the low detail images of Airplane, Couple, Camera, Boats

and Lady images. So, Method-I shows better performance for low detail images.

02: X-ray images are a class of low detail images and Method-I reduces bit rate for all of the images.

03: Images with smooth areas can be encoded efficiently, because smooth areas can produce consecutive zeros in the error image and consecutive same errors can be encoded by single code using run length.

04: Number of symbols in the code table is large for images with lot of edges and less for images with low number of edges, like X-ray images.

05: Medical images are normally low detail images, so expecting to work good Method-I for these images.

06. Large size images have less overhead per pixel compared to small size images.

3.7 Conclusion

A new method of error modeling for lossless image compression has been developed, implemented and tested in this chapter. This method can be coupled with any types of prediction technique. Results show that by incorporating this method with any type of prediction can improve the bit rate or compression ratio. So, this method can be an option of lossless image compression where time is not a prime factor.

CHAPTER 4

METHOD-II: A SYMMETRIC TECHNIQUE FOR LOSSLESS IMAGE COMPRESSION

4.1 Introduction

This chapter describes Method-II in details. In lossless JPEG, decoding time is much higher than encoding time. This large difference between encoding and decoding time makes it unattractive for real time applications. For real time application what the encoder is sending within a particular time decoder must recover information in the same time. If there is a delay between encoding and decoding, it can't be applicable for real time. Method-II in this thesis will attempt to reduce the decoding time by implementing a small code table. The main objective of this method is to make a symmetric technique where decorrelation is done by predictive method. A technique is called symmetric when the difference between encoding and decoding time is

small and is called asymmetric when this difference is high. In real time applications the decoder should finish decoding an image at the same time the encoder does.

The range of the errors in any DPCM technique for 8 bit image can vary from –255 to 255, which gives a total of 511 symbols. In standard JPEG, the prediction errors are generated first and based on the frequency of occurrence every prediction error is given a unique Huffman code. In Method-II prediction errors will be generated in the same manner as in lossless JPEG, but every symbol will not be assigned a unique Huffman code. Instead, Method-II will group the prediction errors based on the 'useful' most significant bit position of their absolute values and assign a Huffman code for the group. Here emphasis is given to the word 'useful'. The useful most significant bit position means that the position of the first occurrence of '1' in the eight bit binary word when checking bits from the most significant bit to the least significant bit. In an eight bit precision image, Method-II will divide the prediction errors into eight groups and each group is given a Huffman code. These Huffman codes consists of variable number of codewords. So, the number of codes in the code table will be eight. Lossless JPEG or any other DPCM technique will search a code table which consists of a large number of codes (Max 511) to decode any symbol. But Method-II will

search a small code table that consists of only eight codes. So, the searching time to decode any image is greatly reduced in Method-II but the cost to be paid is in the compression ratio.

Khalid Sayood and Karen Anderson have investigated a low complexity lossless image compression algorithm which is suitable for real-time application [42]. But the cost paid by their method in terms of compression ratio is high. They did not use any prediction based technique. In their algorithm, they divided the pixel values in terms of suffix and prefix by comparing the current eight bits with the previous eight bits. Method-II will develop a prediction based symmetric technique which will provide a faster encoding and decoding while maintaining a good compression ratio.

4.2 Grouping Of The Prediction Errors

In DPCM technique the prediction error can be any one from the possible 511 symbols. These prediction errors are concentrated around the zero mean in the histogram. The range of the absolute values of the prediction errors can vary from 0 to 255. To represent any prediction error, unsigned eight bits are necessary to represent its absolute value and an additional one bit is needed to represent its sign. In most of the cases, absolute values can be represented by the first four or five least

significant bits because the prediction errors are packed around zero. Negative numbers are stored in the computer as a 2's complement. An example of how the prediction errors are stored and how its absolute values look like is shown below.

Prediction error	Binary representation	Absolute value in binary
-5	1111 1011	0000 0101
2	0000 0010	0000 0010
–1	1111 1111	0000 0001
1	0000 0001	0000 0001
0	0000 0000	0000 0000
3	0000 0011	0000 0011
-3	1111 1101	0000 0011
18	0001 0010	0001 0010

From the above example it is apparent that the ones in the absolute values of the prediction errors are packed in the first four or five least significant bits. The grouping is based on the position of the useful most significant bits. The position of these useful most significant bits can be any of the eight possible bits. So, eight groups are made based on this position.

The possible eight groups are as follows:

Group 1: Least significant bit will determine its value and others are zero

0	0000 000**0**	Least significant bit is 0.
1	0000 000**1**	Least significant bit is 1.

Group 2: 2nd least significant bit is always 1 and 3rd to 8th bits are zero

2	0000 00**1**0	2nd Least significant bit is 1.
3	0000 00**1**1	2nd Least significant bit is 1.

Group 3: 3rd least significant bit is always 1 and 4nd to 8th bits are zero

4	0000 0**1**00	3rd Least significant bit is 1.
5	0000 0**1**01	3rd Least significant bit is 1.
6	0000 0**1**10	3rd Least significant bit is 1.
7	0000 0**1**11	3rd Least significant bit is 1.

Group 4: 4th least significant bit is always 1 and 5th to 8th bits are zero

8	0000 **1**000	4th Least significant bit is 1.
9	0000 **1**001	4th Least significant bit is 1.
10	0000 **1**010	4th Least significant bit is 1.
11	0000 **1**011	4th Least significant bit is 1.
12	0000 **1**100	4th Least significant bit is 1.
13	0000 **1**101	4th Least significant bit is 1.
14	0000 **1**110	4th Least significant bit is 1.
15	0000 **1**111	4th Least significant bit is 1.

Similarly

Group 5: 5th bit will be always 1 and 6th to 8th bit will be zero.

Group 6: 6th bit will be always 1 and 7th to 8th bit will be zero.

Group 7: 7th bit will be always 1 and 8th bit will be always zero.

Group 8: 8th bit will be always 1.

In summary

Group 1: Element 0, 1

Group 2: Element 2,3

Group 3: Element 4 to 7

Group 4: Element 8 to 15

Group 5: Element 16 to 31

Group 6: Element 32 to 63

Group 7: Element 64 to 127

Group 8: Element 128 to 255

Due to the nature of the error image, almost 90 % of the prediction errors will fall within group 1 to group 4.

4.3 Encoding

Statistics have to be taken for different groups to encode the prediction errors. Statistics will indicate the distribution of the prediction errors within the different groups. Figure 4.1 shows how a typical distribution

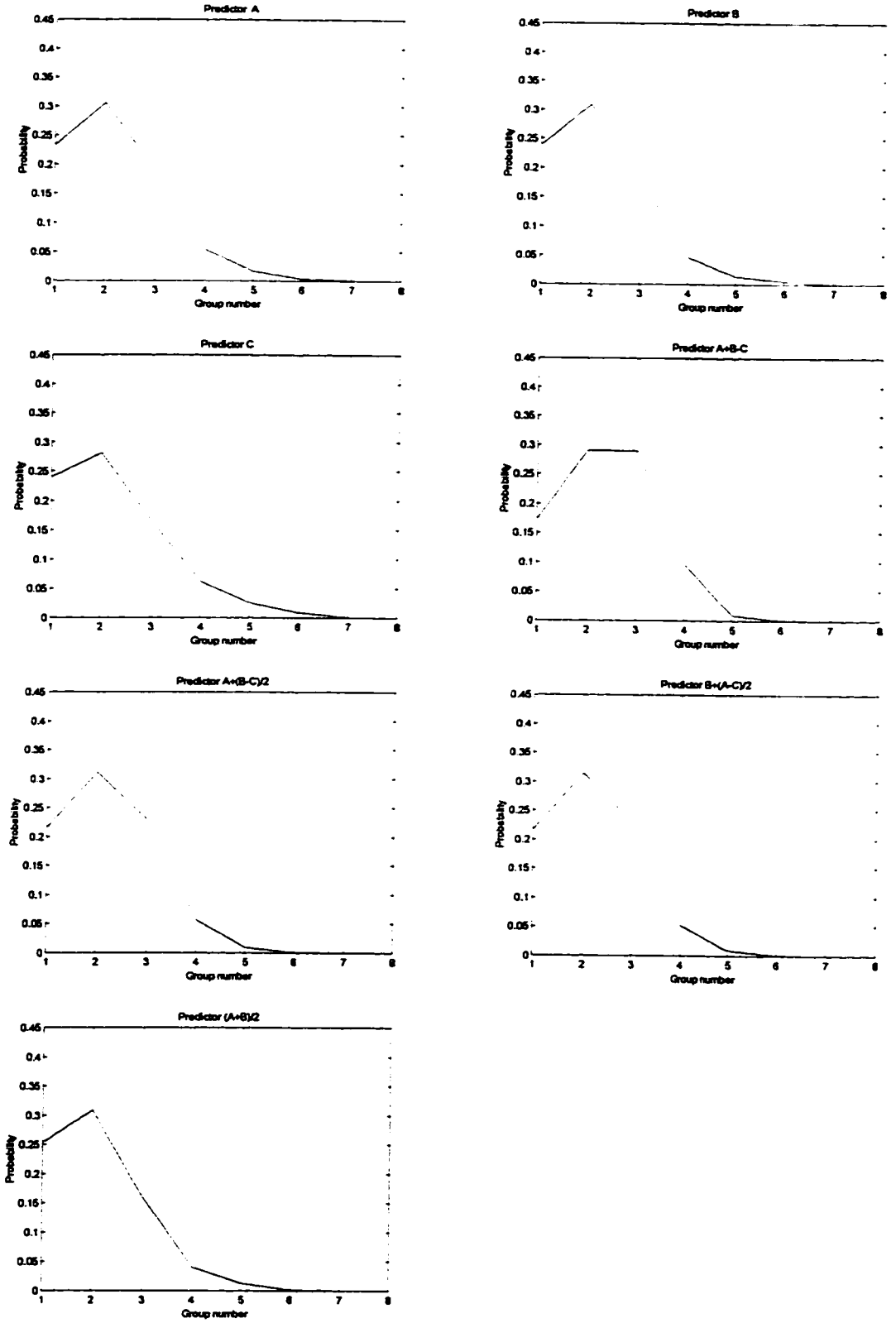


Figure 4.1: Probability distribution of groups in a typical image Peppers

of prediction errors for different groups looks like. These typical distributions are taken from test image peppers using seven JPEG predictors. From the Figure 4.1 it can be observed that the shape of the distribution is almost same for the different predictors which indicates the robustness of the distribution with different predictors.

Huffman codes are assigned to different groups before encoding. These huffman codes will represent the group number of any prediction error. The group number will represent the useful most significant bit position and the bit in that position is '1' except the first group. Whenever the useful most significant bit position is identified then the remaining bits of the absolute values of the prediction errors to be sent as it is and one sign bit will follow the remaining bits.

Figure 4.2 shows the block diagram of the compression process of Method-II. This block diagram has two stages: 'prediction' and 'group coder'.

The first block in the block diagram is called prediction block. This block will generate prediction errors using JPEG predictors. JPEG predictors are given in Table 2.2 of chapter two. The second block, 'group coder', will determine the group of the prediction error and will select the

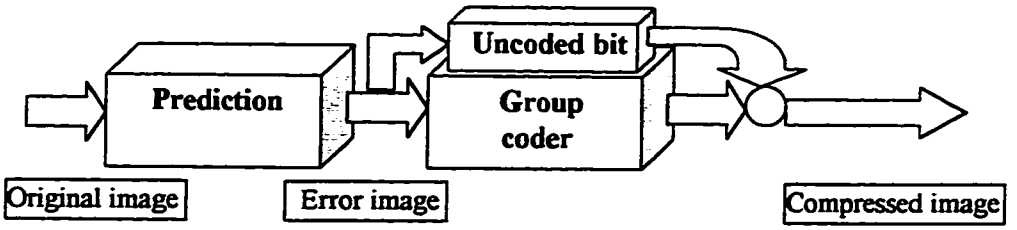


Figure 4.2: Block diagram of compression process by Method-II

code for the group from the code table. To determine the group of any prediction error, we must first convert the prediction error to its absolute value then find out its group. All prediction errors will fall within any of the eight possible groups.

To complete the encoding of any prediction error: first, encode the code for the group; second, encode the remaining bits of the absolute value of prediction errors; third, a sign bit to be coded if the prediction error is other than zero. The number of remaining bits can be calculated by the following equation:

$$F = \begin{cases} G-1 & \text{if } 1 < G \le 8 \\ 1 & \text{if } G = 1 \end{cases} \qquad (4.1)$$

Where F is the number of remaining bits which will follow the group code and G is the group number. Figure 4.3 shows the complete code pattern of a typical prediction error.

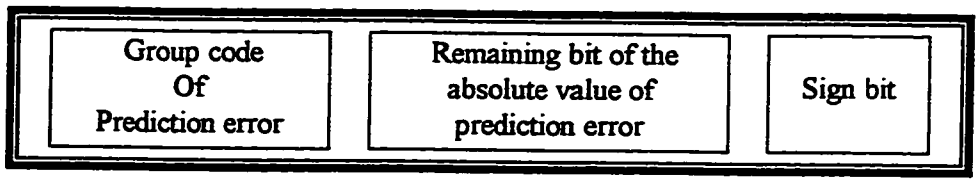


Figure 4.3: Complete code pattern of a typical prediction error

Encoding example

An arbitrary prediction error 13 is chosen for encoding. Its actual value and absolute value is the same. This number 13 will fall in group 4. Binary representation of 13 is 0 0 0 0 1 1 0 1. If the Huffman code for group 4 is 001 and the sign bit '1' represents a positive value and '0' represents a negative value. The complete code for prediction error 13 will be 001 101 1.

There is an exception for the first group. For the first group, the least significant bit of the absolute values of prediction errors will follow the group code and if the least significant bit is zero then no need to code the sign bit because the prediction error is zero. There is a significant number of prediction errors with zero value which will not required any sign bit.

In summary

Code for group 1 will be followed by 1 bit as a remaining bit which is the least significant bit and a single bit for sign if the error is not zero

Similarly

Code for group 2 will follow 1 bit and 1 sign bit

Code for group 3 will follow 2 bit and 1 sign bit

Code for group 4 will follow 3 bit and 1 sign bit

Code for group 5 will follow 4 bit and 1 sign bit

Code for group 6 will follow 5 bit and 1 sign bit

Code for group 7 will follow 6 bit and 1 sign bit

Code for group 8 will follow 7 bit and 1 sign bit

Most of the prediction errors will fall within the first four groups, so the number of bits needed to encode any prediction error within these groups are less compared to the last four groups.

4.4 Decoding

Figure 4.4 shows the block diagram of decompression process by Method-II. Decoding in the first block is carried out in two stages. In the first stage, the group is identified by decoding the group code and this is

done by searching in the code table. After decoding group code, the remaining bits of the prediction error are calculated using equation 4.1, which is F. So, the next F bits are decoded as is from the bit stream without searching the code table. After decoding the remaining bits, prediction error is checked whether it is zero or not. If the prediction error is not zero check for the next bit which is the sign bit.

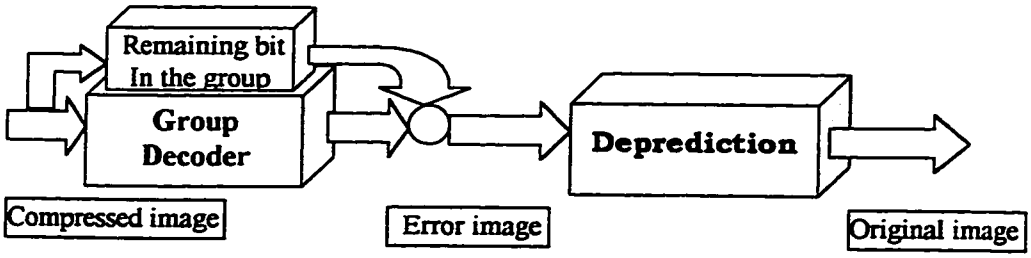


Figure 4.4: Block diagram of decompression process by Method-II

The output of the first block are fed to the second block where original pixel values are generated from prediction errors. In this block original pixel values are recovered by adding the prediction errors with the predicted values.

Decoding example

An arbitrary bit stream 0011011 is chosen for decoding. The memory location of the prediction error is initialized to 0000 0000 before decoding. In the first stage of decoding, the code table is searched to find

a match between bit steam and any code in the code table. Suppose a match is found with the code of group 4 which is 001. The group number 4 indicates that the fourth least significant bit is '1'. So, insert '1' to the fourth least significant bit position. In this stage prediction error will be 0000 1000.

By the group number 4 and using equation 4.1 the remaining (4-1) = 3 bits to be copied from the bit stream to the location after the fourth least significant bit. From the bit stream the remaining three bits are detected as 101. After copying these three bits, the prediction error will be 0000 1101. Now check the prediction whether it is zero or not. The prediction error recovered in this stage is not zero for this example. So, check the next bit in the bit stream to detect the sign of the prediction error. In this example, the next bit in the bit stream is 1. This '1' indicates that the sign of the prediction error is positive. So, no need to change the sign of the prediction error because it is already positive. Finally the recovered prediction error is 0000 1101 which is 13.

In the last block, the prediction errors are added with the predicted value to form the original pixel values with no loss at all.

4.5 Results

In the following sections, simulation results are presented for Method-II. The test images used in the simulations are taken from different classes. The performance of Method-II is compared with lossless JPEG in terms of the compression ratio, encoding speed and decoding speed. Tables and graphs are presented to visualize these performances.

Every image is coded based on its own distribution of groups. A summary of the results of the test images using the best predictor is presented at the end of this chapter. A best predictor means the predictor which gives the highest compression ratio among all predictors. The percentage of bits in the bit stream encoded by a Huffman coder for different images is also presented in a tabular and a graphical form. The total number of bits in the code table for JPEG and Method-II is given in a table.

In the following sections, encoding time means the time required to encode the error image from the prediction errors. The time required for generation of prediction errors is not considered as a part of the encoding time in this work because this time is same for both methods. The decoding time is the time required to recover the prediction errors from the compressed bit stream. Time for generation of the original pixel values from the prediction errors is not considered as a part of the

decoding time because this time is also the same for both methods. So, the time under investigation by Method-II is the time which is algorithm dependent. From the block diagram of Figure 4.2, the time for the first block 'prediction' and from the block diagram of Figure 4.4, the time for the last block 'Deprediction' is not considered as a part of encoding and decoding time respectively.

The encoding and decoding times are compared by ratio rather than by difference. The encoding and decoding times are machine dependent. So, the difference between encoding and decoding time will also be machine dependent. To avoid this dependency, the performance is compared in terms of ratios. These ratios indicate how many times Method-II is faster when compared to JPEG. A ratio 1 indicates that the encoding and decoding time is same for both methods. A ratio greater than 1 indicates that Method-II is faster and a ratio smaller than 1 indicates that Method-II is slower. These results can be converted to a percentile by multiplying these ratios by 100.

In the following sections, simulation results of the test image "Lady" is presented. The results for the other images almost convey the same message. So results for other images are presented in tabular form in appendix C and results of the two representative images 'Lena' and 'Boats' are presented in graphical form in appendix D. Finally, results summary are presented at the end of this chapter.

4.5.1 Simulation Results of test image lady

Lady (256pixel x 256 pixel)

Simulation results of lady image are presented in Table 4.1. The encoding and the decoding time required by both methods are presented in Figure 4.5(a) and the magnified version of the lower part of Figure 4.5(a) is shown in Figure 4.5(b). From the Figure 4.5(b) it is clear that the encoding time by JPEG varies within .25 seconds. From Table 4.1, the minimum encoding time is 0.16 seconds for predictor 5,6,7 and the maximum encoding time is 0.22 seconds for predictor 1,2,3. The decoding time for JPEG varies from 11 to 15 seconds. The minimum decoding time is 11.81 seconds for predictor 7 and the maximum is 14.17 seconds for predictor 3.

The encoding time varies within 0.2 seconds by Method-II. The minimum is .16 second for predictor 3,4,5,6 and the maximum is .17 seconds for predictor 1,2,7. When compared with JPEG, Method-II takes less encoding time for six predictors out of seven. The encoding speed of Method-II varies from 1.0625 times to 1.375 times faster than JPEG. But one out of seven predictors takes more encoding time than JPEG. The speed of Method-II for that predictor is .941 times that of JPEG which indicates that encoding speed of Method-II is slower than JPEG. On the average Method-II is 1.138 times faster than JPEG.

Table 4.1: Simulation results of lady image by Method-II

Lady Predictors	L-JPEG				Method - II				Comparison			
	Bit-rate Bits/pel	Compr Ratio	Encoding Time (sec)	Decoding Time (sec)	Bit-rate Bits/pel	Compr Ratio	Encoding Time (sec)	Decoding Time (sec)	Diff in Bit rate Bits/pel (M-II-JPEG)	Chang In Comp. Ratio (%)	Ratio of Encoding Time (JPEG/M-II)	Ratio of Decoding Time (JPEG/M-II)
A	4.8256	1.6578	0.22	12.96	4.9127	1.6284	0.17	0.33	0.0871	-1.77	1.29	39.27
B	4.7889	1.6705	0.22	12.91	4.8852	1.6376	0.17	0.33	0.0963	-1.97	1.29	39.12
C	5.1809	1.5441	0.22	14.17	5.2499	1.5238	0.16	0.39	0.069	-1.31	1.38	36.33
A+B-C	4.6643	1.7152	0.17	12.35	4.7719	1.6765	0.16	0.33	0.1076	-2.26	1.06	37.42
A+(B-C)/2	4.544	1.7606	0.16	11.92	4.6834	1.7082	0.16	0.28	0.1394	-2.98	1.00	42.57
B+(A-C)/2	4.5299	1.7660	0.16	11.92	4.6713	1.7126	0.16	0.28	0.1414	-3.02	1.00	42.57
(A+B)/2	4.4973	1.7788	0.16	11.81	4.6498	1.7205	0.17	0.33	0.1527	-3.28	0.94	35.78
Average	4.7187	1.6954	0.19	12.58	4.8320	1.6556	0.16	0.32	0.1134	-2.35	1.14	39.01

L-JPEG : Lossless JPEG

M-II : Method-II

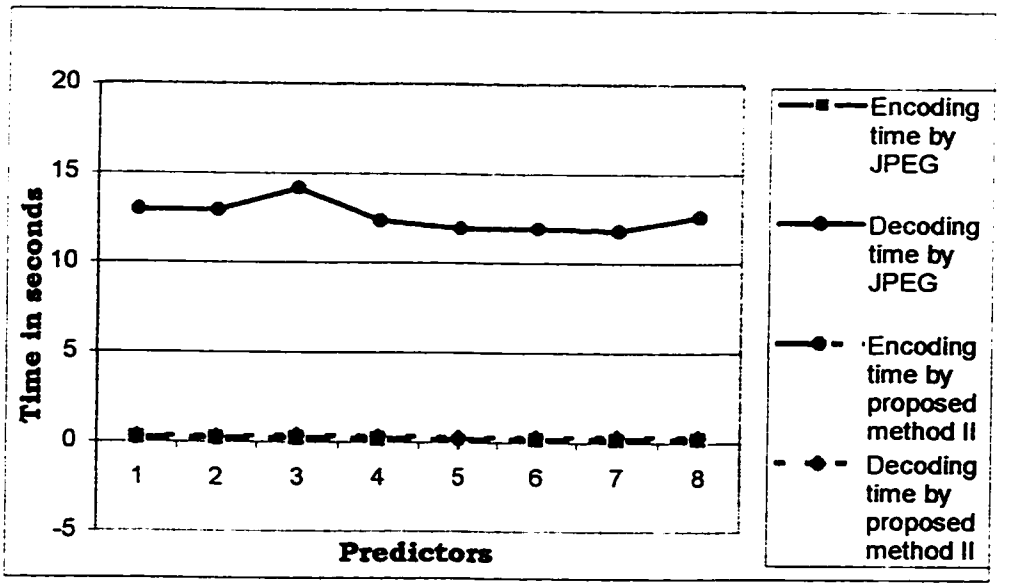


Figure 4.5(a) : Encoding and decoding time for lady

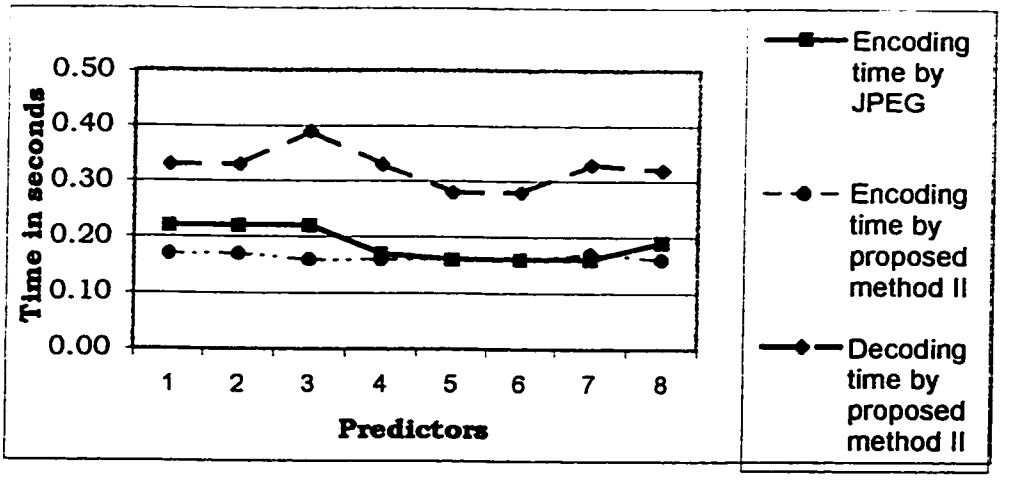


Figure 4.5(b): Magnified view of bottom part of Figure 4.5(a)

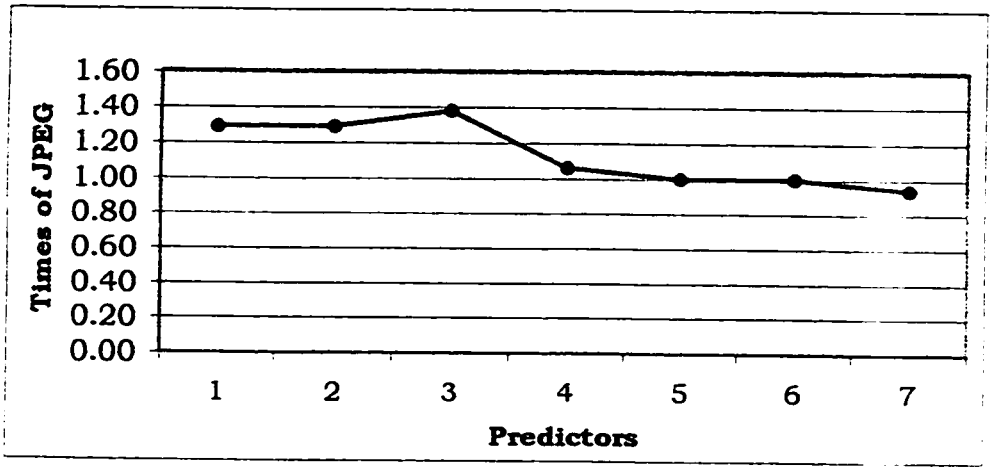


Figure 4.5(c): Encoding speed of Method-II over JPEG for lady

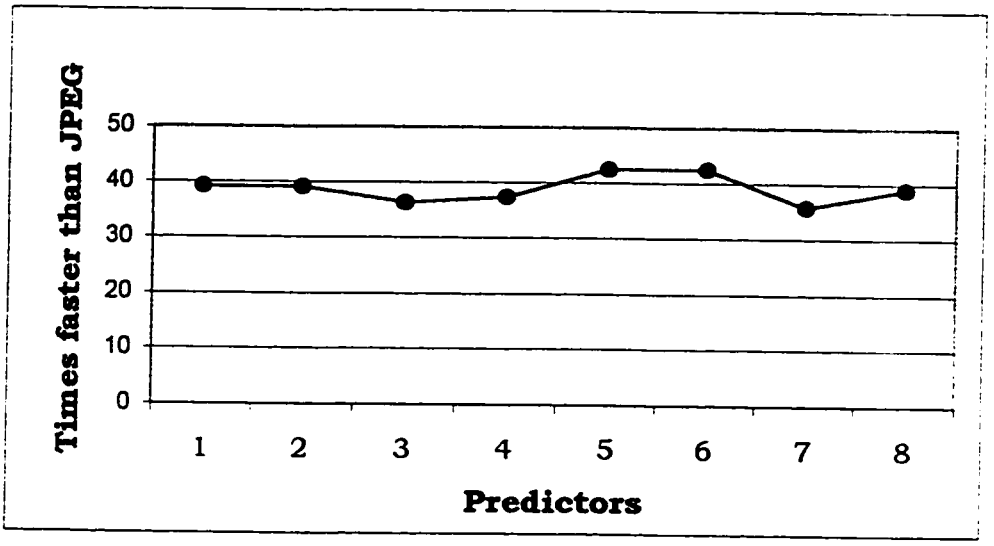


Figure 4.5(d) : Decoding speed of Method-II over JPEG for lady

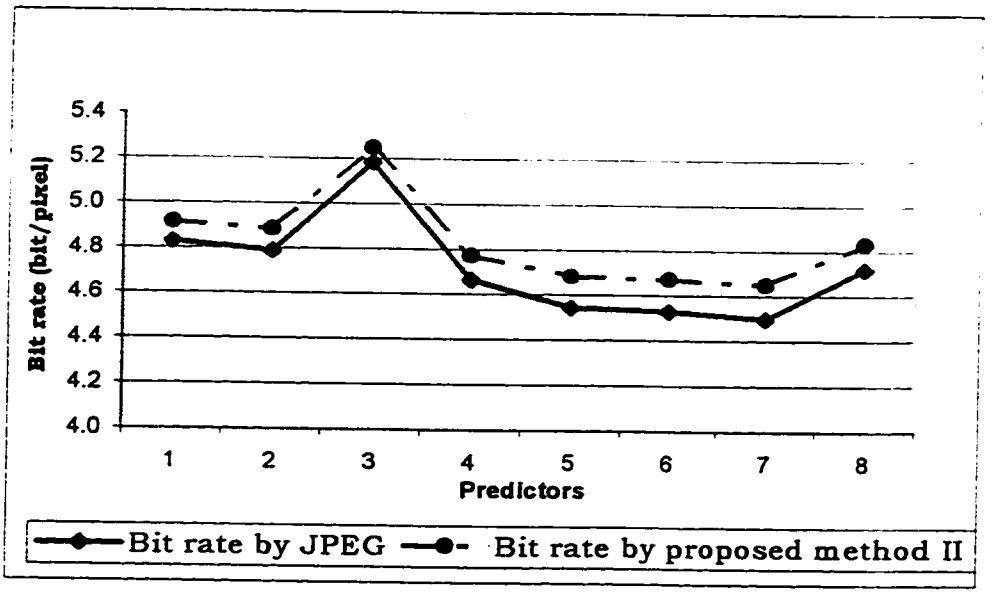


Figure 4.5(e): Bit rate comparison for lady

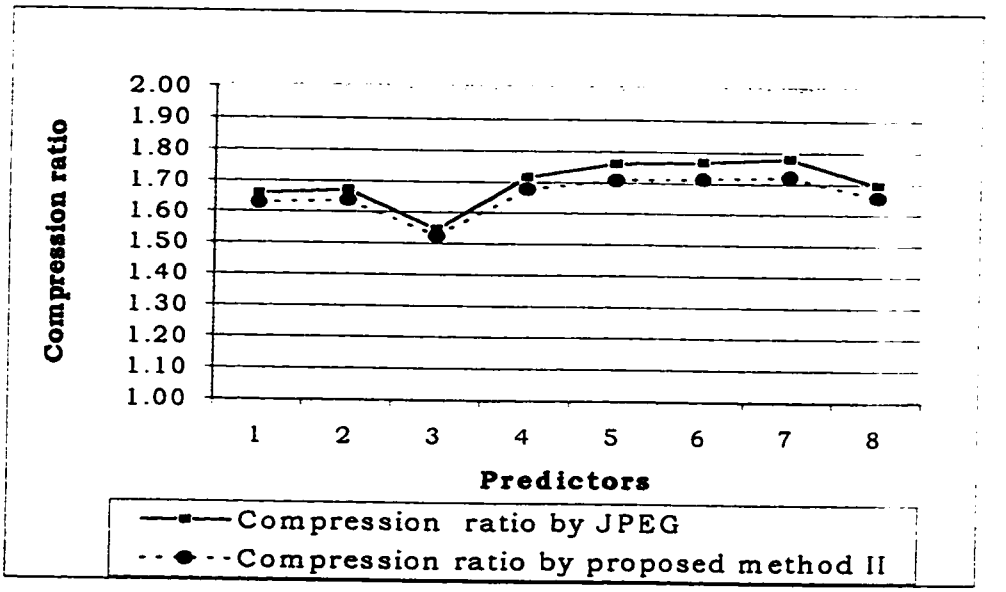


Figure 4.5(f): Compression ratio comparison for lady

The big reduction in decoding time using Method-II is clearly visible from the Figure 4.5(a). The minimum improvement in the decoding speed is 35.78 times and the maximum is 42.57 times when compared to JPEG. On the average, the decoding speed of Method-II is 39.01 times faster. So it is vividly a big gain in terms of decoding time.

The encoding and decoding speed of Method-II when compared to JPEG is presented in Figure 4.5(c) and 4.5(d). From Figure 4.5(c) a small gain in encoding speed can be seen. Figure 4.5(d) clearly shows a big gain in the decoding speed by Method-II. The difference between the average encoding and average decoding time is 12.39 sec by JPEG and 0.16 sec by Method-II. So, Method-II clearly reduces the difference between encoding and decoding time and makes the system symmetric.

Figure 4.5(e) shows the bit rate achieved by JPEG and Method-II. The compression ratio by JPEG and Method-II is shown in Figure 4.5(f). There are two curves for each of these figures - one for JPEG and one for Method-II. The curves for Method-II lies above JPEG. It indicates that the bit rate by Method-II is more than the bit rate by JPEG. So, there is a penalty in bit rate by Method-II when compared to JPEG. The same thing is presented in Figure 4.5(f) in terms of compression ratio. The average compression ratio decreased by Method-II is 2.35%. Figure 4.5(f) shows

that both curves are close to each other which indicates very small loss of compression ratio by Method-II.

The difference between the average encoding and the average decoding time is 12.39 seconds by JPEG and 0.16 seconds by Method-II. Compared to the small loss in compression ratio with the small gain in encoding speed and the big gain in decoding speed by Method-II, it is easy to conclude that the speed improvement can overrides the loss in compression ratio.

4.5.2 Results Summary

The improvement in speed using Method-II is mainly due to the uncoded bits in the bit stream and the lower number of bits in the code table. More than 50% bits are sent directly using Method-II without coding which saves time.

Table 4.2 presents the percentage of coded and uncoded bits for the different images using the best predictor. The best predictor is the predictor, which gives the highest compression ratio among the different predictors. The maximum uncoded bit is 58.68% for the baboon image and the minimum is 52.34% for the camera image. On the average, the uncoded bit is 54.16% and the coded bit is 45.84%. The graphical representation of the coded and uncoded bits for the different images are shown in Figure 4.6.

Table 4.2: Percentage of coded and uncoded bits for test images

Image	Lady	Couple	Camera	Bridge	Lena	Baboon	Peppers	Airplane	Boats	Goldhill	Average
% of bit coded by Huffman	47.54	47.20	47.66	42.92	45.59	41.72	45.51	47.10	47.32	45.85	45.84
% of bit sent directly	52.46	52.80	52.34	57.08	54.41	58.68	54.49	52.9	52.68	54.15	54.16
Total	100 %	100 %	100 %	100 %	100 %	100 %	100 %	100 %	100 %	100%	100%

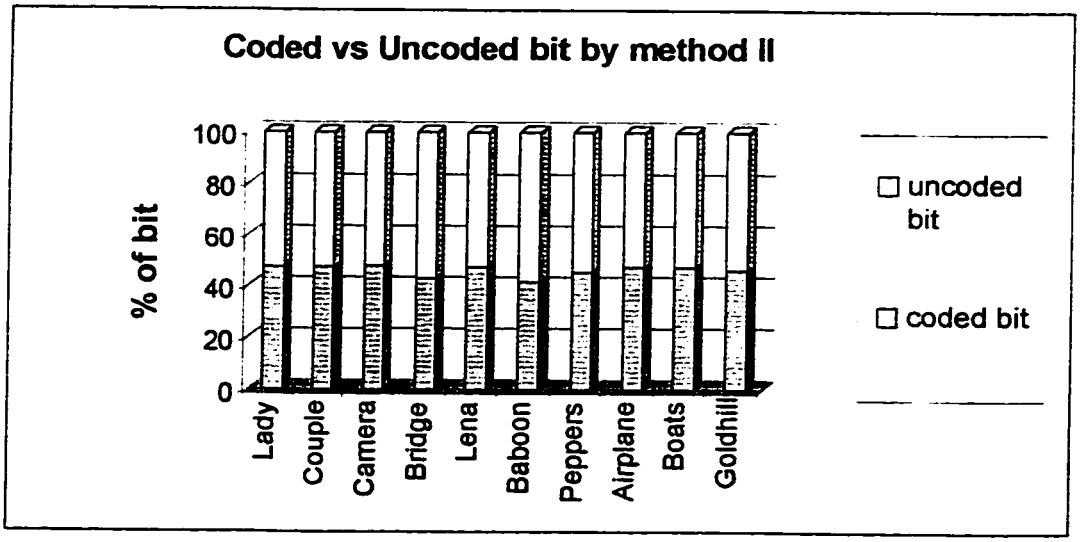


Figure 4.6: Percentage of uncoded and coded bits for test images

The big percentage of uncoded bits shown in Figure 4.6 is one of the main source of saving time. Another source of saving time is the number of bits in the code table for the different images.

Table 4.3: No. of bits in the code table for the test images

	Lady	Couple	Camera	Bridge	Lena	Baboon	Peppers	Airplane	Boats	Goldhill
No of bit in JPEG code table	35	27	35	27	30	27	23	35	30	30
No of bit in Method II code table	1796	1852	3231	2496	2373	2744	2309	1914	2278	4590

Table 4.3 shows the number of bits in the code table for the different images by JPEG and Method-II. From the Table 4.3, it can be observed that the number of bits in the code table using Method-II are tremendously reduced compared to JPEG. So, the time for searching to decode any bit by Method-II is greatly reduced when compared to JPEG. The time saving by the two sources contributes to the improvement of encoding and decoding speed by Method-II.

A summary of the results for the different images using the best predictor are presented in Table 4.4. The performance of Method-II in terms of encoding and decoding speed when compared to JPEG is shown in Figure 4.7 and 4.8 respectively. The encoding speed by Method-II increases for seven images and decreases for three images but on the average gives an improvement by Method-II is 1.19 times compared to JPEG. So, there is a small improvement in encoding speed by Method-II.

Figure 4.8 shows that the decoding speed by Method-II improves in all images and the minimum improvement is 16 times for the test image boats and the maximum is 43.03 times for the baboon image when compared to JPEG. On the average, the decoding speed improvement by Method-II is 33.17 times compared to JPEG.

The bit rate achieved by Method-II and JPEG using the best predictor for different images is shown in Figure 4.9. The curve for

Table 4.4: Results summary of the test images using best predictor

Test Image	L-JPEG				Method- II				Comparison			
	Bit-rate Bits/pel	Compr Ratio	Encoding Time (sec)	Decoding Time (sec)	Bit-rate Bits/pel	Compr Ratio	Encoding Time (sec)	Decoding Time (sec)	Diff in Bit rate Bits/pel (M-II-JPEG)	Chang In Comp. Ratio (%)	Ratio of Encoding Time (JPEG/ M-II)	Ratio of Decoding Time (JPEG/ M-II)
Lady	4.4973	1.7788	0.16	11.75	4.6498	1.7205	0.17	0.33	.1525	-3.28	0.94	35.60
Couple	4.2096	1.9004	0.16	10.82	4.2786	1.8698	0.11	0.27	.0690	-1.61	1.45	40.07
Camera	4.9326	1.6219	0.22	13.46	5.0674	1.5787	0.17	0.38	.1348	-2.66	1.29	35.42
Bridge	6.0404	1.3244	0.33	17.30	6.1622	1.2982	0.22	0.43	.1218	-1.98	1.50	40.23
Lena	4.9595	1.6131	0.77	54.16	5.0755	1.5762	0.66	2.03	.1160	-2.29	1.17	26.67
Baboon	6.2982	1.2702	1.26	73.16	6.4439	1.2415	0.83	1.70	.1457	-2.26	1.52	43.03
Peppers	4.8919	1.6353	0.77	53.39	5.0210	1.5933	0.88	1.37	.1291	-2.57	0.88	38.97
Airplane	4.3565	1.8364	0.77	45.32	4.4390	1.8022	0.88	1.21	.0825	-1.86	0.88	37.45
Boats	4.5058	1.7755	1.10	75.85	4.6525	1.7195	0.98	4.73	.1467	-3.15	1.12	16.00
Goldhill	5.1368	1.5574	1.31	91.34	5.2172	1.5334	1.10	5.00	.0804	-1.54	1.19	18.27
Average	4.9829	1.6313	0.69	44.66	5.1007	1.5933	0.60	1.75	.1179	-2.33	1.19	33.17

L-JPEG : Lossless JPEG

M-II : Method II

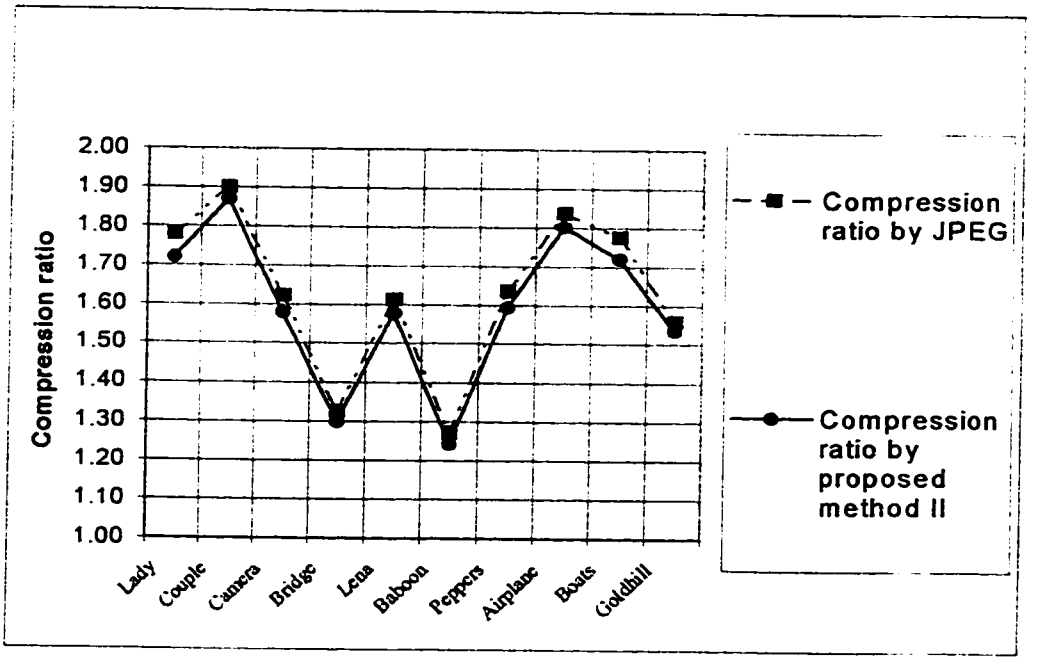


Figure 4.7: Encoding speed of Method-II using best predictor

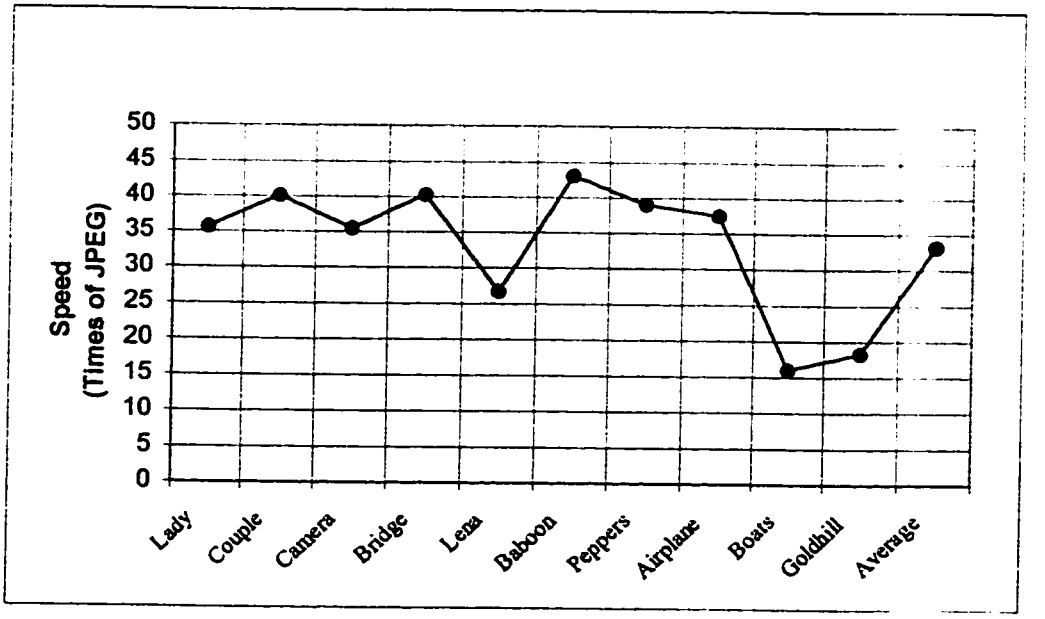


Figure 4.8: Decoding speed of Method-II using best predictor

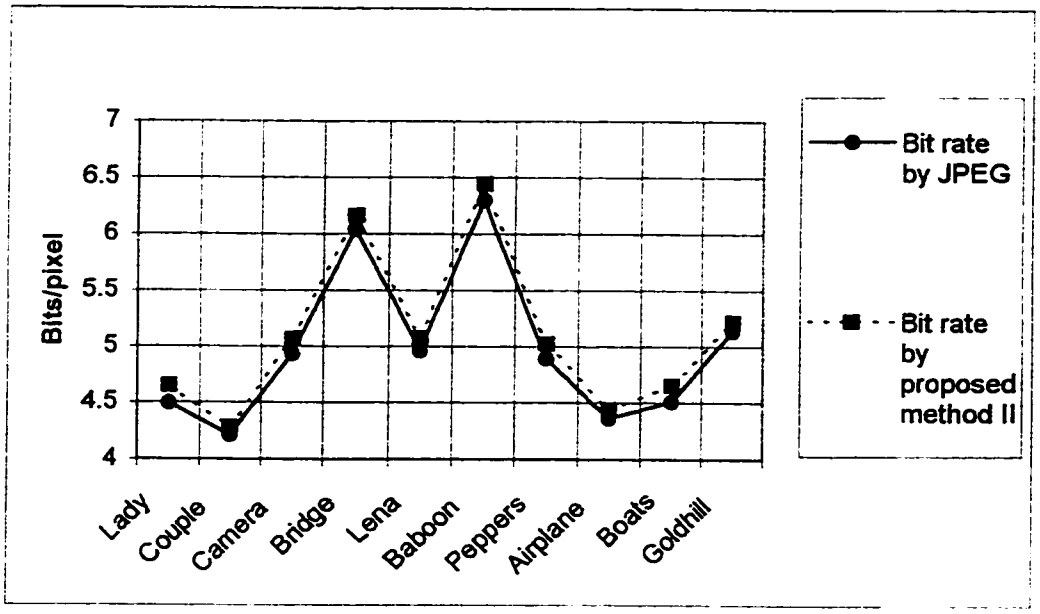


Figure 4.9: Bit rate of test images using best predictor

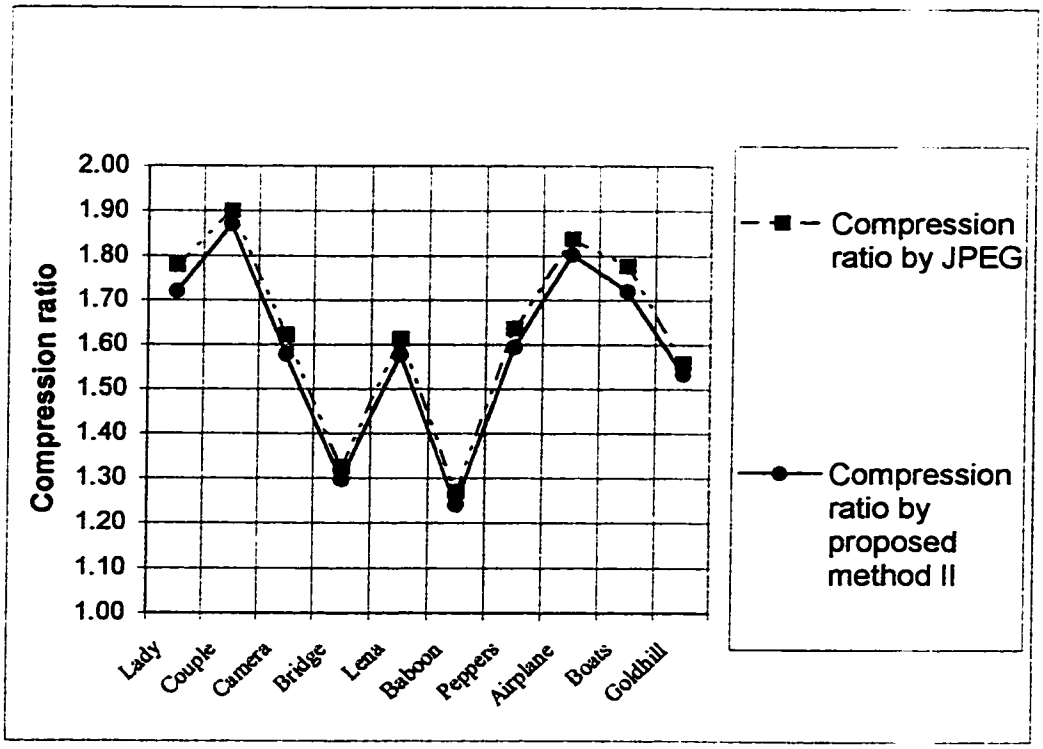


Figure 4.10: Compression ratio of test images using best predictor

Method-II is always lies above JPEG which indicates that the bit rate is increased by Method-II. The maximum increase in bit rate by Method-II is .1525 bits/pixel for lady image and the minimum is .069 bits/pixel for couple image. The average bit rate loss by Method-II to increase the decoding speed is .1179 bits/pixel.

Figure 4.10 shows the compression ratio achieved by Method-II and JPEG. This curves are the inverse of the curves shown in Figure 4.9. The maximum loss in compression ratio by Method-II is 3.28% and the minimum is 1.54%. The average compression ratio sacrificed by Method-II is 2.33%. The difference in the average encoding and the average decoding time is 43.97 seconds by JPEG and is 1.15 seconds by Method-II.

So, Method-II reduces the difference between encoding and decoding time and makes the system symmetric at the cost of small compression ratio.

4.6 Conclusion

A low complexity encoding method for lossless image compression has been developed, implemented and tested in this chapter. Method-II is a new symmetric technique which can be coupled with any types of prediction to improve the encoding and decoding speed. This method achieves high decoding speed at the cost of small compression ratio. So, this method can be a practical option for real time application.

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusion

The major contribution of this thesis is the development of two new methods for lossless image compression for two areas of application. Method-I can be applied where reduction in bit rate is the main target and Method-II can be applied for real time application where encoding and decoding time is a factor. Both methods are implemented and tested in the domain of visual C++ programming language.

Improvement in compression ratio is achieved by modeling the prediction errors in Method-I. Method-I maps the prediction errors to a binary bit stream. Bit streams are divided into groups and each group is encoded based on its own code table. The ratio of the number-of-zeros to the number-of-ones in the different groups of the bit streams are

different. So, Method-I takes the advantage of different concentration of ones, measured by the ratio of number-of-zeros to the number-of-ones, in different groups of the bit stream. Two groups of results have been presented for Method-I. The first group of images are selected from different classes and the second group of images are selected from the same class. The followings are the conclusions drawn from the study of Method-I

- Method-I is an effective error modeling technique for lossless image compression
- Any sub method of Method-I can improve the bit rate but MG-M (Multi-Group-Mix) can be the best choice.
- Even though the prediction is done by JPEG predictors but any types of prediction will work.
- Method-I can be coupled with any types of prediction technique where errors are generated by prediction and prediction errors are entropy coded. So, Method-I can be used as an practical option for error modeling. It is up to the user to choose the error modeling or not. If the option is chosen for 'modeling', Method-I will work. If the option is chosen for 'no modeling', Method-I is bypassed and the prediction errors are encoded by the entropy coder. Figure 5.1 describes how the algorithms will choose.

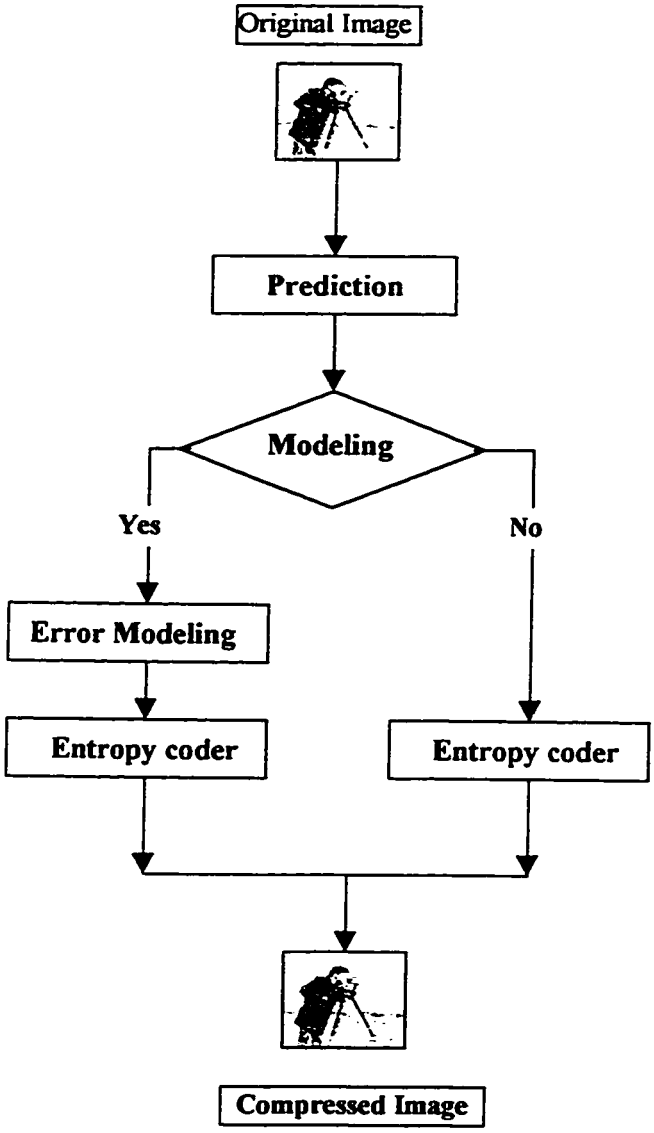


Figure 5.1: Block diagram showing Method-I as an option

- The improvement in bit rate by Method-I is achieved at the cost of encoding and decoding time as well as memory requirement.

- Method-I is not a practical option when time is a critical factor.

Method-II is a fast and symmetric lossless image compression technique. This method encodes the prediction errors using small code table. Due to the small size code table, this method takes less time to decode any symbol. This method is tested on images from different classes and the following conclusion can be made:

- Method-II is a fast and symmetric technique for lossless image compression.
- It reduces the difference between encoding and decoding time which is a criteria for real time transmission.
- The loss in compression ratio by Method-II is small.
- Even though results are generated by JPEG predictors, it can be used with any types of prediction method.
- This method shows the same results in images from different classes. So this method is suitable for images from any class.

5.2 Recommendations For Future Work

The following recommendations for future work are made to improve the performance and verify the validity of the two methods developed in this thesis:

Method-I

Even though results are generated using JPEG predictors, the results can be verified by more simulation using different types of predictors.

Bit streams of Method-I can be coded using more efficient binary coder like JBIG instead of Huffman coder.

Different groups of bit stream can be coded using different coders. As an example, the first group can be coded using arithmetic coder [29], the second group can be coded by JBIG [19] and the third group can be coded by logical coding [20].

Attempt can be possible to reduce the complexity by separating sign and absolute values of prediction errors and encode them separately.

Method-II

By taking statistics from a large number of images a standard code table can be made and the validity of Method-II can be verified using the standard code table.

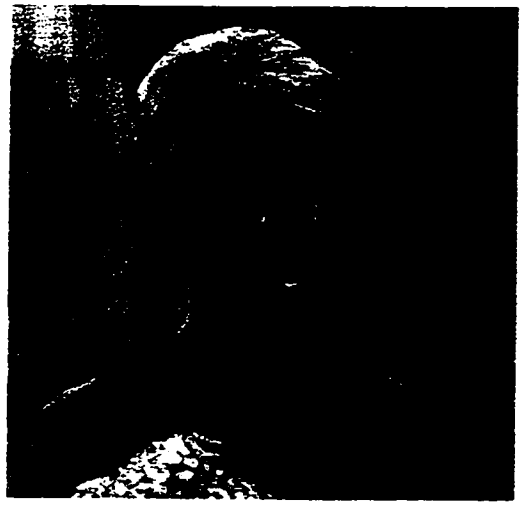
Attempt can be made to use some other coders to encode the remaining bits instead of sending directly.

This algorithm can be implemented in hardware as a tool for real time application.

Appendix A

Test images
of
Group 1

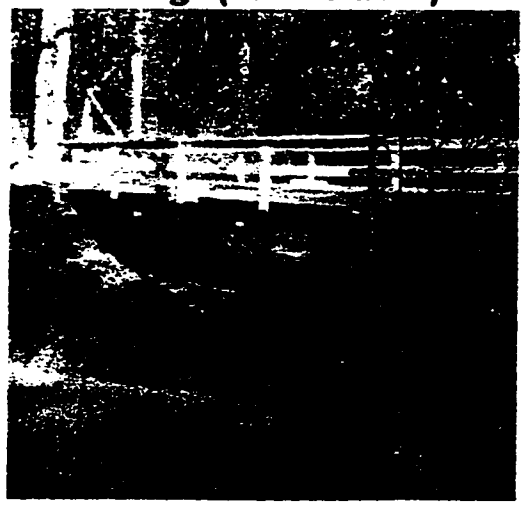
Lady (256 x 256)



Couple (256 x 256)

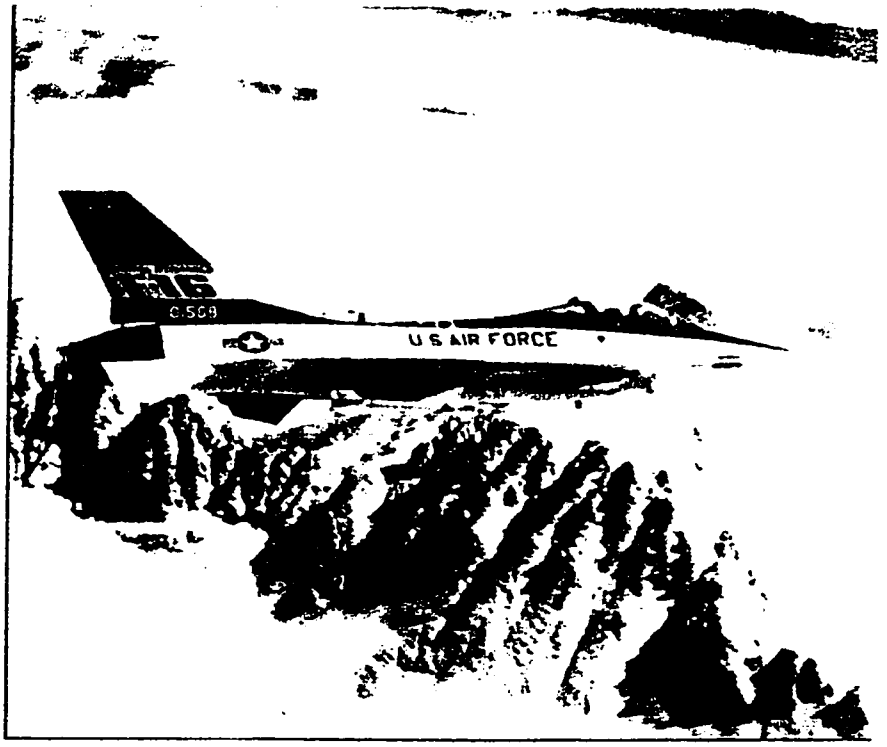
Camera (256 x 256)

Bridge (256 x 256)



Baboon (512 x 512)

Airplane (512 x 512)



Peppers (512 x 512)

Boats (576 x 720)

Goldhill (576 x 720)

Appendix B

Test images
of
Group 2 (X-ray)

X-ray 1

X-ray 2

X-ray 3

X-ray 4

Appendix C

Results of Method-II
(Tabular representation)

Table C.1: Simulation results of couple image by Method-II

Couple / Predictors	L-JPEG				Proposed Method II				Comparison			
	Bit-rate Bits/pel	Compr Ratio	Encoding Time (sec)	Decoding Time (sec)	Bit-rate Bit/pel	Compr Ratio	Encoding Time (sec)	Decoding Time (sec)	Diff in Bit rate Bit/pel (PM-JPEG)	Chang In Comp. Ratio (%))	Ratio of Encoding Time (JPEG/PM)	Ratio of Decoding Time (JPEG/PM)
A	4.7186	1.6954	0.22	12.69	4.8135	1.6620	0.16	0.33	0.0949	-1.97	1.38	38.45
B	4.5017	1.7771	0.16	11.92	4.5839	1.7452	0.17	0.33	0.0822	-1.80	0.94	36.12
C	5.1947	1.5400	0.22	11.44	5.2838	1.5141	0.16	0.33	0.0891	-1.68	1.38	34.66
A+B-C	4.2679	1.8745	0.22	10.98	4.3448	1.8413	0.16	0.33	0.0469	-1.77	1.38	33.27
A+(B-C)/2	4.3414	1.8427	0.16	11.26	4.4223	1.8090	0.16	0.33	0.0809	-1.83	1.00	34.12
B+(A-C)/2	4.2096	1.9004	0.16	10.98	4.2786	1.8698	0.11	0.28	0.069	-1.61	1.45	39.21
(A+B)/2	4.4305	1.8057	0.22	11.54	4.5212	1.7694	0.16	0.28	0.0907	-2.01	1.38	41.21
Average	4.5235	1.7685	0.19	11.54	4.6069	1.7365	0.15	0.32	0.0791	-1.81	1.27	36.72

L-JPEG : Lossless JPEG

PM : Proposed method II

Table C.2: Simulation results of camera image by Method-II

Camera / Predictors	L-JPEG				Proposed Method II				Comparison			
	Bit-rate Bits/pel	Compr Ratio	Encoding Time (sec)	Decoding Time (sec)	Bit-rate Bit/pel	Compr Ratio	Encoding Time (sec)	Decoding Time (sec)	Diff in Bit rate Bit/pel (PM-JPEG)	Chang In Comp. Ratio (%))	Ratio of Encoding Time (JPEG/PM)	Ratio of Decoding Time (JPEG/PM)
A	5.0787	1.5752	0.22	14.12	5.2128	1.5347	0.17	0.32	0.1341	-2.57	1.29	44.13
B	5.1072	1.5664	0.27	14.12	5.2304	1.5295	0.16	0.33	0.1232	-2.36	1.69	42.78
C	5.4988	1.4549	0.28	15.76	5.5841	1.4326	0.17	0.38	0.0853	-1.53	1.65	41.78
A+B-C	5.1187	1.5629	0.27	14.07	5.1995	1.5386	0.16	0.39	0.0808	-1.55	1.68	36.07
A+(B-C)/2	4.9628	1.6120	0.22	13.40	5.0947	1.5703	0.17	0.38	0.1319	-2.59	1.29	35.26
B+(A-C)/2	4.978	1.6071	0.22	13.63	5.1043	1.5673	0.17	0.38	0.1263	-2.48	1.29	35.86
(A+B)/2	4.9326	1.6219	0.22	13.40	5.0674	1.5787	0.17	0.38	0.1348	-2.66	1.29	35.26
Average	5.093	1.5708	0.24	14.07	5.2156	1.5339	0.17	0.36	0.1226	-2.35	1.42	39.18

L-JPEG : Lossless JPEG

PM : Proposed method II

Table C.3: Simulation results of bridge image by Method-II

Bridge / Predictors	L-JPEG				Proposed Method II				Comparison			
	Bit-rate Bit/pel	Compr Ratio	Encoding Time (sec)	Decoding Time (sec)	Bit-rate Bit/pel	Compr Ratio	Encoding Time (sec)	Decoding Time (sec)	Diff in Bit rate Bit/pel (PM-JPEG)	Chang In Comp. Ratio (%)	Ratio of Encoding Time (JPEG/PM)	Ratio of Decoding Time (JPEG/PM)
A	6.1997	1.2904	0.33	18.07	6.3057	1.2687	0.22	0.44	0.106	-1.68	1.50	41.06
B	6.4239	1.2453	0.33	19.01	6.5306	1.2250	0.22	0.44	0.1067	-1.63	1.50	43.20
C	6.7292	1.1888	0.38	20.10	6.8031	1.1759	0.17	0.49	0.0739	-1.09	2.23	41.02
A+B-C	6.2309	1.2839	0.27	18.07	6.3600	1.2579	0.16	0.44	0.1291	-2.03	1.69	41.06
A+(B-C)/2	6.0404	1.3244	0.33	17.25	6.1622	1.2982	0.22	0.39	0.1218	-1.98	1.50	44.23
B+(A-C)/2	6.1507	1.3007	0.33	17.74	6.2852	1.2728	0.17	0.44	0.1345	-2.14	1.94	40.31
(A+B)/2	6.0637	1.3193	0.33	17.35	6.1914	1.2921	0.22	0.39	0.1277	-2.06	1.50	44.48
Average	6.2626	1.2774	0.33	18.23	6.3769	1.2545	0.20	0.43	0.1142	-1.79	1.69	42.19

L-JPEG : Lossless JPEG

PM : Proposed Method II

Table C.4: Simulation results of lena image by Method-II

Lena	L-JPEG				Proposed Method II				Comparison			
Predictors	Bit-rate Bits/pel	Compr Ratio	Encoding Time (sec)	Decoding Time (sec)	Bit-rate Bit/pel	Compr Ratio	Encoding Time (sec)	Decoding Time (sec)	Diff in Bit rate Bit/pel (PM-JPEG)	Chang In Comp. Ratio (%)	Ratio of Encoding Time (JPEG/PM)	Ratio of Decoding Time (JPEG/PM)
A	5.3652	1.4911	1.32	60.20	5.4768	1.4607	0.66	1.54	-.1116	-2.04	2.00	39.09
B	5.0439	1.5861	1.27	55.69	5.1394	1.5566	0.82	3.52	-.0955	-1.86	1.55	15.82
C	5.5787	1.4340	1.04	62.79	5.6743	1.4099	0.77	2.97	-.0958	-1.68	1.35	21.14
A+B-C	5.2026	1.5377	0.99	57.51	5.3230	1.5029	0.82	1.65	-.1204	-2.26	1.21	38.85
A+(B-C)/2	5.0872	1.5726	0.88	55.58	5.2155	1.5339	1.10	2.03	-.1283	-2.46	0.80	27.37
B+(A-C)/2	4.9595	1.6131	0.77	53.94	5.0755	1.5762	0.66	1.37	-.1160	-2.29	1.12	39.37
(A+B)/2	4.9698	1.6097	0.88	54.37	5.0642	1.5797	0.66	1.43	-.0944	-1.86	1.33	38.02
Average	5.1724	1.5467	1.02	57.15	5.2812	1.5148	0.78	2.07	-.1089	-2.06	1.34	31.38

L-JPEG : Lossless JPEG

PM : Proposed Method II

Table C.5: Simulation results of baboon image by Method-II

Baboon / Predictors	L-JPEG				Proposed Method II				Comparison			
	Bit-rate Bits/pel	Compr Ratio	Encoding Time (sec)	Decoding Time (sec)	Bit-rate Bit/pel	Compr Ratio	Encoding Time (sec)	Decoding Time (sec)	Diff in Bit rate Bit/pel (PM-JPEG)	Chang In Comp. Ratio (%)	Ratio of Encoding Time (JPEG/PM)	Ratio of Decoding Time (JPEG/PM)
A	6.3889	1.2522	1.31	75.64	6.5166	1.2276	0.77	2.69	-.1277	-1.96	1.70	28.11
B	6.7363	1.1876	1.53	88.64	6.8497	1.1679	0.94	3.57	-.1134	-1.66	1.62	24.83
C	6.8848	1.1620	1.70	83.71	6.9863	1.1451	0.88	1.87	-.1015	-1.45	1.93	44.76
A+B-C	6.6437	1.2041	1.48	78.60	6.7417	1.1866	0.82	2.58	-.0980	-1.45	1.80	30.46
A+(B-C)/2	6.3616	1.2575	1.43	76.29	6.4978	1.2312	1.21	3.18	-.1382	-2.09	1.18	23.99
B+(A-C)/2	6.4845	1.2337	1.38	76.29	6.6137	1.2096	1.32	3.02	-.1292	-1.95	1.04	25.26
(A+B)/2	6.2982	1.2702	1.26	74.81	6.4439	1.2415	0.83	2.47	-.1457	-2.26	1.51	30.28
Average	6.5426	1.2228	1.44	79.14	6.6642	1.2004	0.97	2.77	-.122	-1.83	1.54	29.67

L-JPEG : Lossless JPEG

PM : Proposed Method II

Table C.6: Simulation results of airplane image by Method-II

Airplane	L-JPEG				Proposed Method II				Comparison			
Predictors	Bit-rate Bits/pel	Compr Ratio	Encoding Time (sec)	Decoding Time (sec)	Bit-rate Bit/pel	Compr Ratio	Encoding Time (sec)	Decoding Time (sec)	Diff in Bit rate Bit/pel (PM-JPEG)	Chang In Comp. Ratio (%)	Ratio of Encoding Time (JPEG/PM)	Ratio of Decoding Time (JPEG/PM)
A	4.6604	1.7166	0.82	50.70	4.7646	1.6790	0.61	2.45	0.1042	-2.19	1.34	20.69
B	4.7114	1.6980	0.93	50.92	4.8083	1.6638	0.94	2.53	0.0969	-2.01	0.99	20.12
C	5.1089	1.5659	1.15	58.55	5.1946	1.5401	0.66	1.49	0.0857	-1.65	1.74	39.29
A+B-C	4.4354	1.8037	0.88	47.12	4.5734	1.7492	0.88	2.03	0.1380	-3.02	1.00	23.21
A+(B-C)/2	4.3565	1.8363	0.77	46.74	4.4390	1.8022	0.88	2.47	0.0825	-1.86	0.88	18.92
B+(A-C)/2	4.4114	1.8135	0.94	46.35	4.4879	1.7826	0.77	2.03	0.0765	-1.70	1.22	22.83
(A+B)/2	4.4299	1.8059	0.99	48.06	4.5077	1.7747	0.99	2.03	0.0778	-1.73	1.00	23.67
Average	4.5877	1.7438	0.93	49.78	4.6822	1.7086	0.82	2.15	0.0945	-2.02	1.17	24.10

L-JPEG : Lossless JPEG

PM : Proposed Method II

Table C.7: Simulation results of peppers image by Method-II

Peppers / Predictors	L-JPEG				Proposed Method II				Comparison			
	Bit-rate Bit/pel	Compr Ratio	Encoding Time (sec)	Decoding Time (sec)	Bit-rate Bit/pel	Compr Ratio	Encoding Time (sec)	Decoding Time (sec)	Diff in Bit rate Bit/pel (PM-JPEG)	Chang In Comp. Ratio (%)	Ratio of Encoding Time (JPEG/PM)	Ratio of Decoding Time (JPEG/PM)
A	5.1137	1.5644	0.88	56.79	5.2420	1.5261	0.66	2.69	-.1283	-2.45	1.33	21.11
B	5.0452	1.5857	0.93	54.71	5.1777	1.5451	0.66	1.42	-.1325	-2.56	1.40	38.52
C	5.2197	1.5327	0.88	58.44	5.3040	1.5083	0.99	3.02	-.0843	-1.59	0.89	19.35
A+B-C	5.3869	1.4851	0.93	62.67	5.5211	1.4490	0.71	1.49	-.1342	-2.43	1.30	42.06
A+(B-C)/2	5.1172	1.5634	0.83	57.34	5.2494	1.5240	0.66	1.38	-.1322	-2.52	1.25	41.55
B+(A-C)/2	5.0947	1.5703	0.88	56.57	5.2310	1.5293	1.21	1.87	-.1447	-2.61	0.72	30.25
(A+B)/2	4.8919	1.6354	0.77	52.56	5.0210	1.5933	0.88	3.07	-.1291	-2.57	0.88	17.12
Average	5.1242	1.5612	0.87	57.01	5.2495	1.5240	0.82	2.13	-.1265	-2.38	1.11	29.99

L-JPEG : Lossless JPEG

PM : Proposed Method II

Table C.8: Simulation results of boats image by Method-II

Boats / Predictors	L-JPEG				Proposed Method II				Comparison			
	Bit-rate Bits/pel	Compr Ratio	Encoding Time (sec)	Decoding Time (sec)	Bit-rate Bit/pel	Compr Ratio	Encoding Time (sec)	Decoding Time (sec)	Diff in Bit rate Bit/pel (PM-JPEG)	Chang In Comp. Ratio (%)	Ratio of Encoding Time (JPEG/PM)	Ratio of Decoding Time (JPEG/PM)
A	4.9484	1.6167	1.87	86.12	5.0238	1.5924	0.99	4.94	0.0754	-1.50	1.89	17.43
B	4.7179	1.6957	1.16	82.00	4.8503	1.6494	1.05	6.75	0.1325	-2.73	1.10	12.15
C	5.3115	1.5062	1.49	95.35	5.3911	1.4839	1.10	5.82	0.0796	-1.48	1.35	16.38
A+B-C	4.5772	1.7478	1.16	78.49	4.7129	1.6975	0.99	5.33	0.1357	-2.88	1.17	14.72
A+(B-C)/2	4.5939	1.7414	1.16	79.15	4.7177	1.6957	0.99	4.78	0.1238	-2.62	1.17	16.55
B+(A-C)/2	4.5058	1.7755	1.10	77.72	4.6525	1.7195	0.98	5.33	0.1467	-3.15	1.12	14.58
(A+B)/2	4.6383	1.7248	1.15	80.30	4.7753	1.6753	0.99	5.43	0.1370	-2.87	1.16	14.78
Average	4.7561	1.6821	1.30	82.73	4.8748	1.6411	1.01	5.48	0.1187	-2.44	1.28	15.23

L-JPEG : Lossless JPEG

PM : Proposed Method II

Table C.9: Simulation results of goldhill image by Method-II

Goldhill	L-JPEG				Proposed Method II				Comparison			
Predictors	Bit-rate Bits/pel	Compr Ratio	Encoding Time (sec)	Decoding Time (sec)	Bit-rate Bit/pel	Compr Ratio	Encoding Time (sec)	Decoding Time (sec)	Diff in Bit rate Bit/pel (PM-JPEG)	Chang In Comp. Ratio (%))	Ratio of Encoding Time (JPEG/PM)	Ratio of Decoding Time (JPEG/PM)
A	5.1368	1.5574	1.31	91.34	5.2171	1.5334	1.10	5.22	0.0803	-1.54	1.19	17.49
B	7.8291	1.0218	4.12	163.90	7.9312	1.0087	1.49	5.87	0.1021	-1.28	2.76	27.92
C	7.8310	1.0216	4.17	162.64	7.9342	1.0083	1.43	5.66	0.1032	-1.30	2.91	28.73
A+B-C	5.7743	1.3854	1.65	106.34	5.8855	1.3593	1.15	5.38	0.1112	-1.88	1.43	19.76
A+(B-C)/2	5.3810	1.4867	1.43	96.50	5.4983	1.4550	1.10	5.11	0.1173	-2.13	1.30	18.88
B+(A-C)/2	6.9426	1.1523	2.69	135.01	7.0757	1.1306	1.32	5.66	0.1331	-1.88	2.03	23.85
(A+B)/2	6.8871	1.1616	2.58	135.06	7.0183	1.1399	1.27	6.15	0.1312	-1.87	2.03	21.96
Average	6.5403	1.2232	2.56	127.26	6.6515	1.2027	1.27	5.58	0.1112	-1.68	1.95	22.66

L-JPEG : Lossless JPEG

PM : Proposed Method II

APPENDIX D

RESULTS OF METHOD-II

Graphical representation of test images Lena and Boats

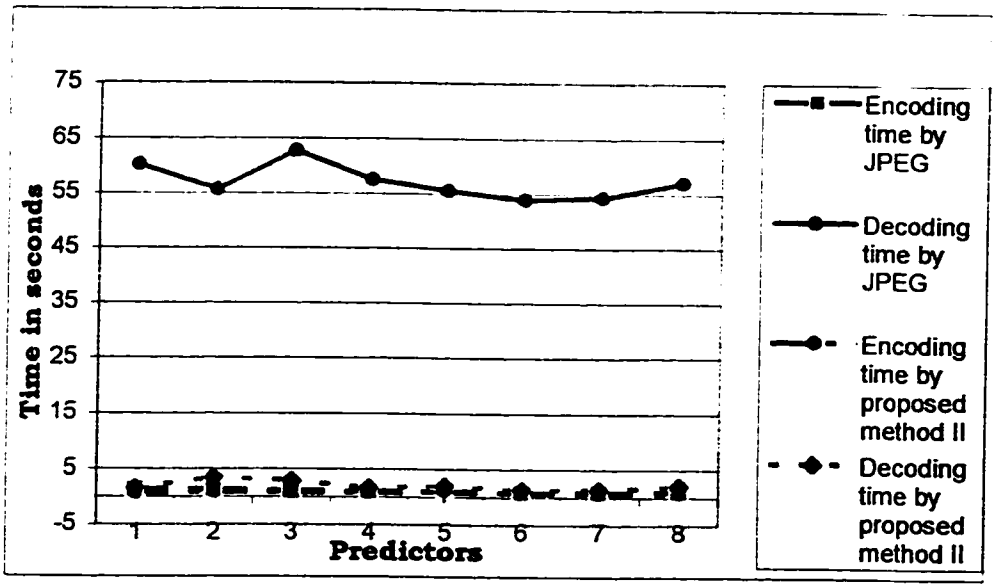


Figure D.1(a): Encoding and decoding time for lena

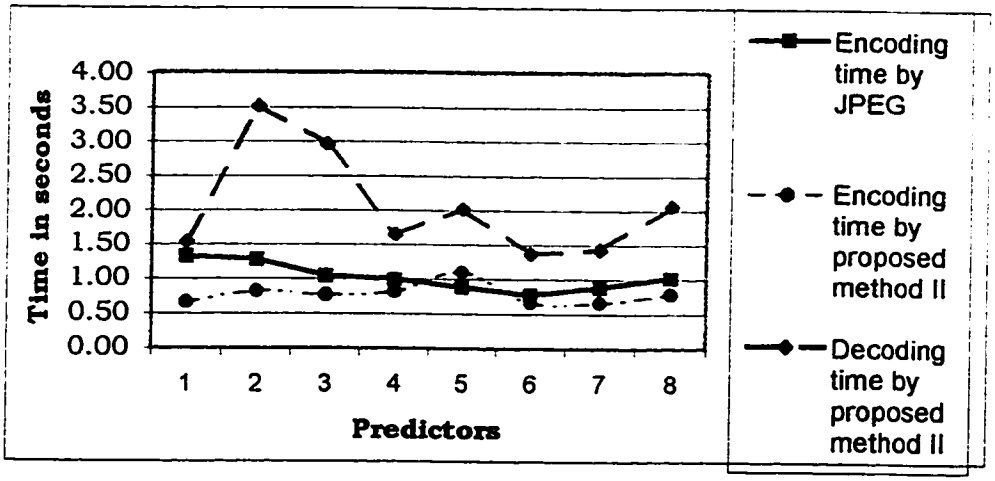


Figure D.1(b) : Magnified view of bottom part of Figure D.1 (a)

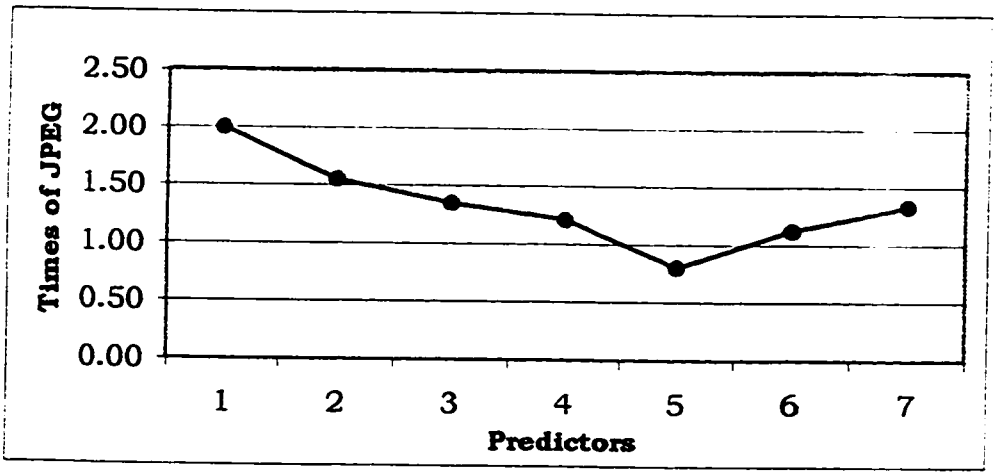


Figure D.1(c): Encoding speed of Method-II over JPEG for lena

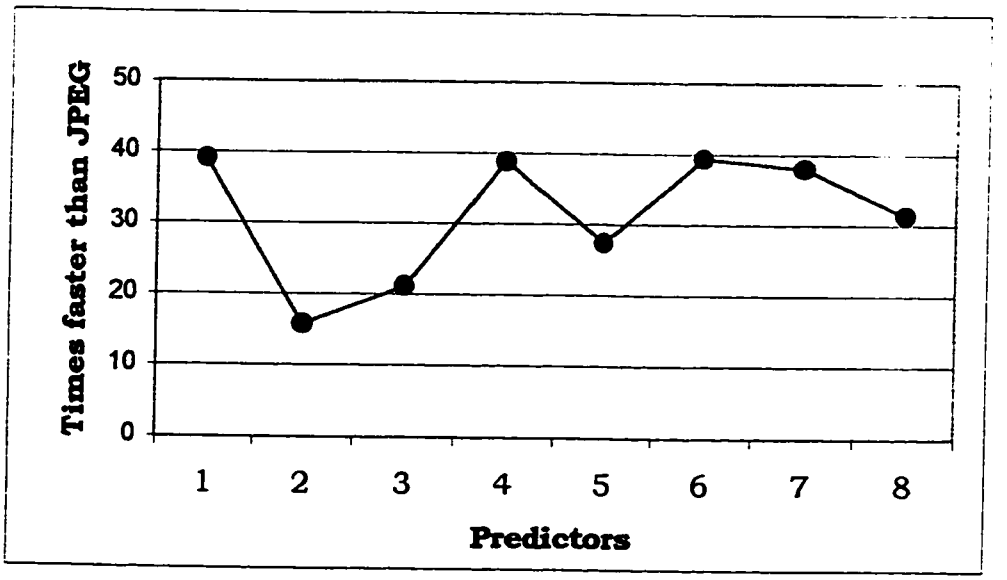


Figure D.1(d): Decoding speed of Method-II over JPEG for lena

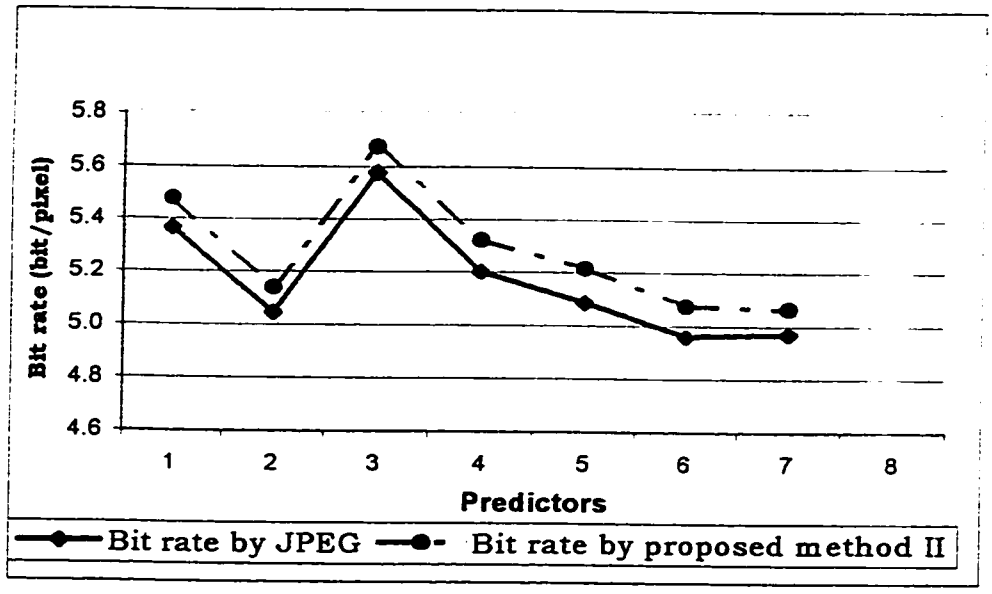


Figure D.1(e): Bit rate comparison of lena

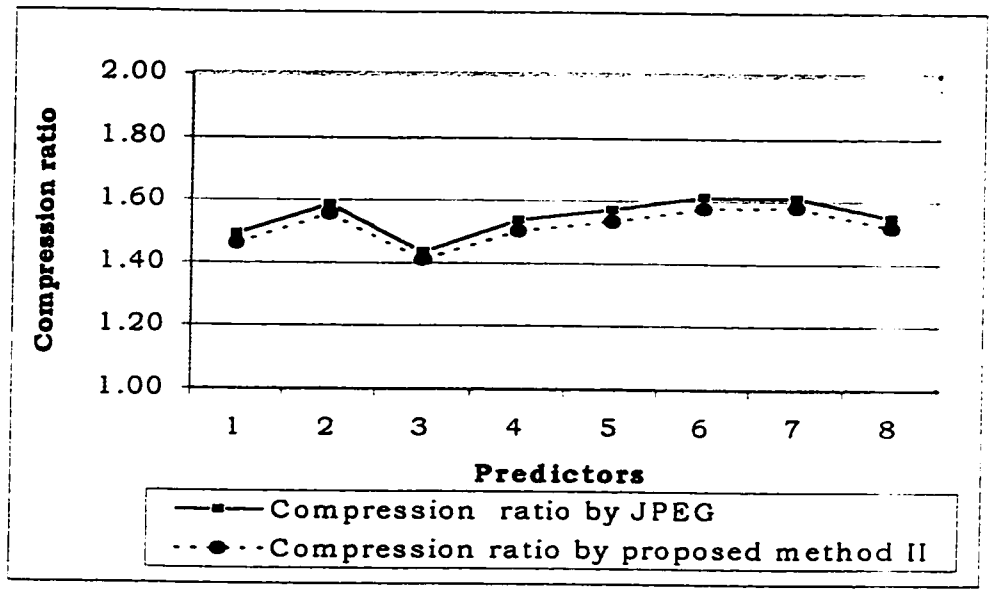


Figure D.1(f): Comparison of compression ratio for lena

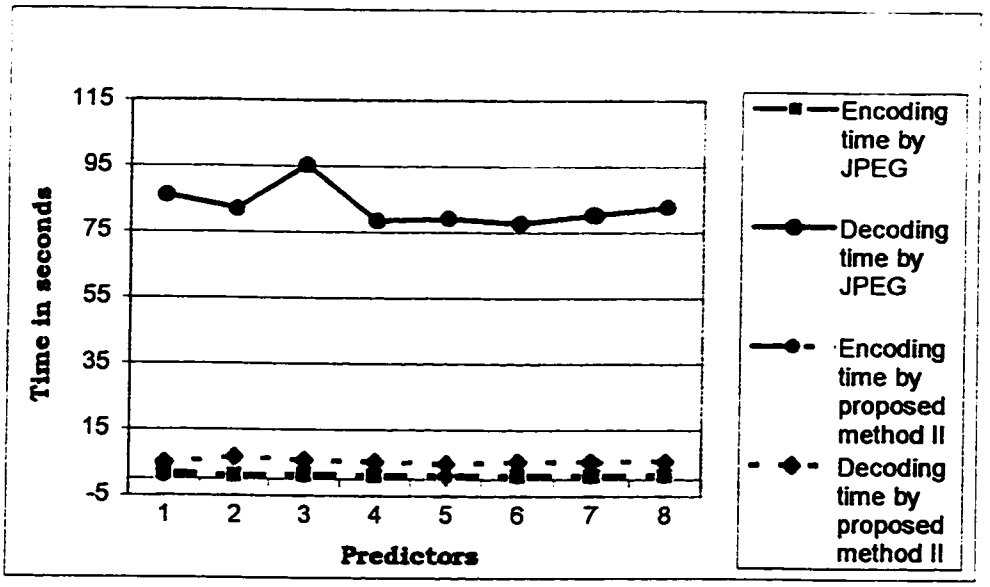


Figure D.2(a) : Encoding and decoding time for boats

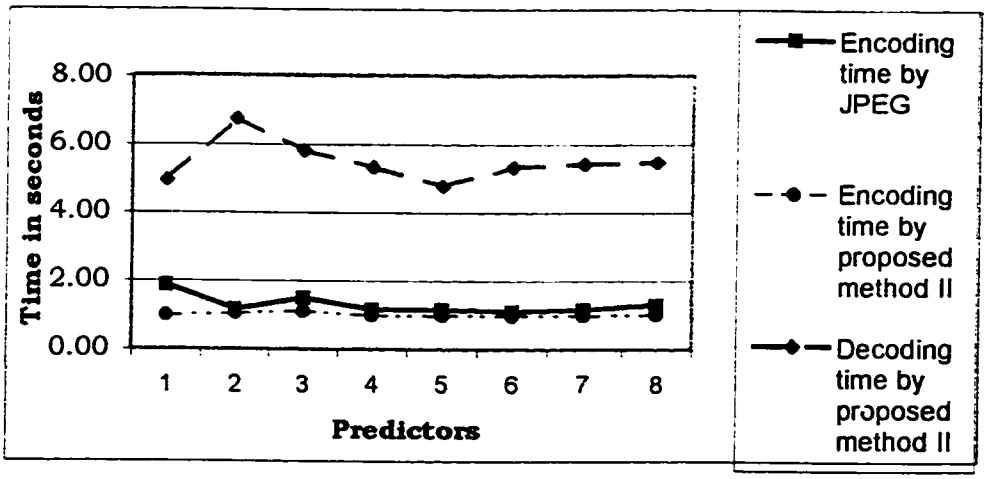


Figure D.2(b) : Magnified view of bottom part of Figure D.2(a)

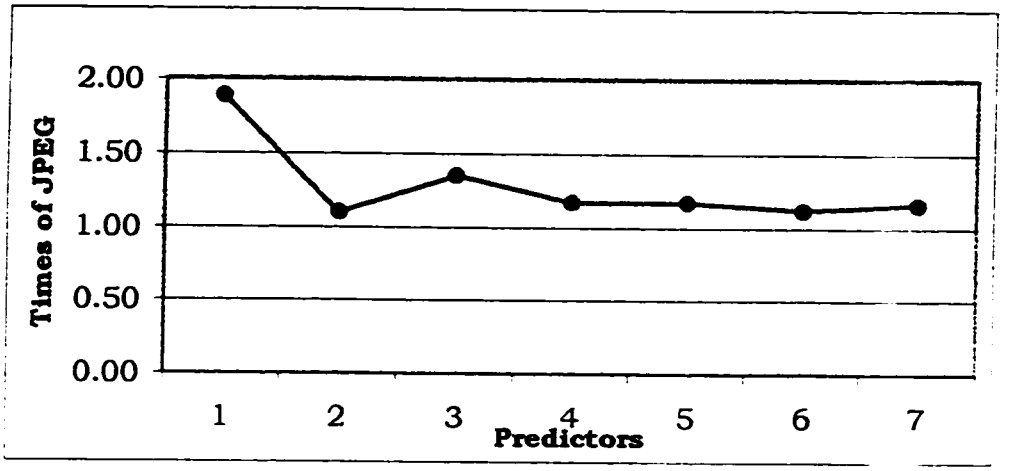


Figure D.2(c): Encoding speed of Method-II over JPEG for boats

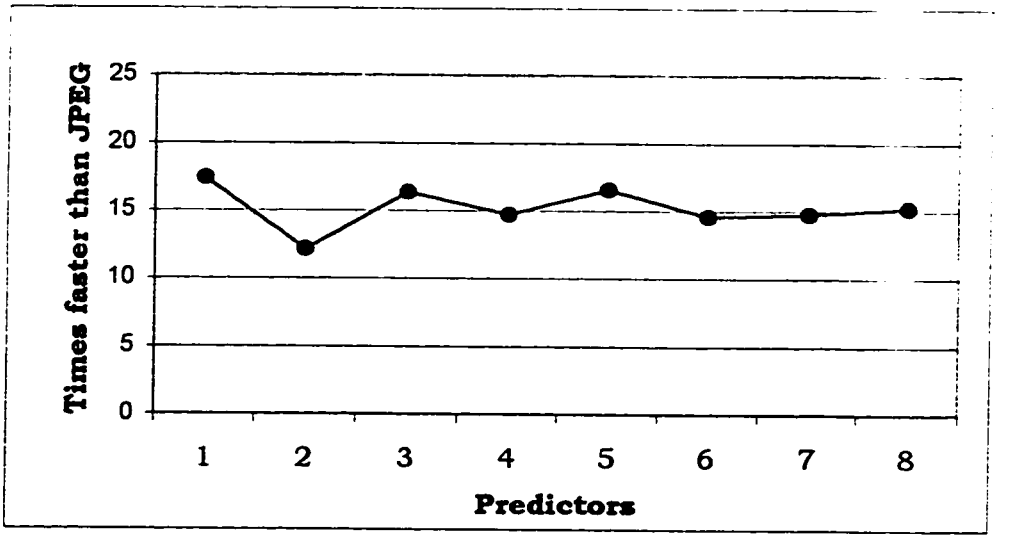


Figure D.2(d): Decoding speed of Method-II over JPEG for boats

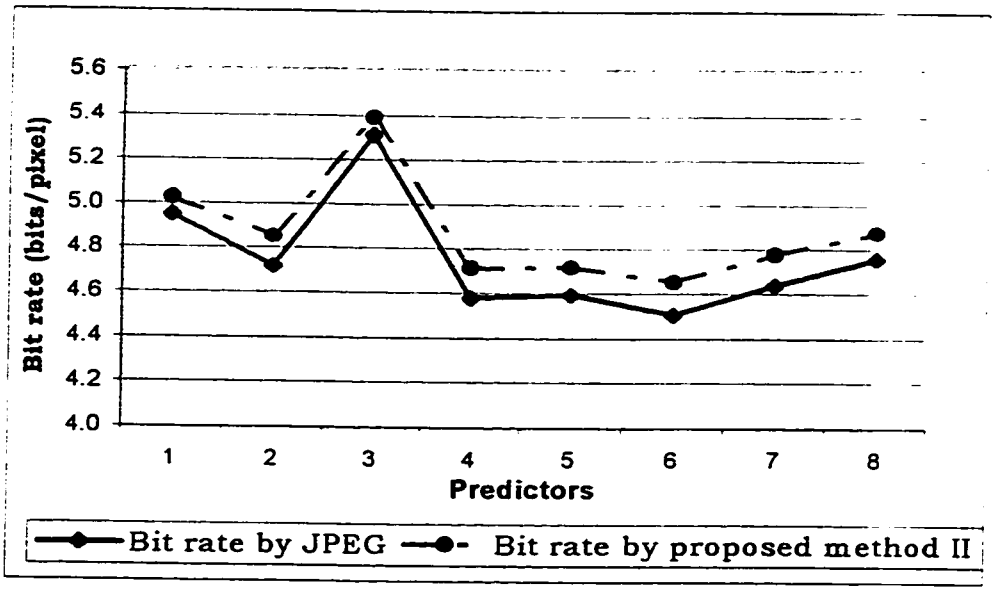


Figure D.2(e): Comparison of Bit rate for boats

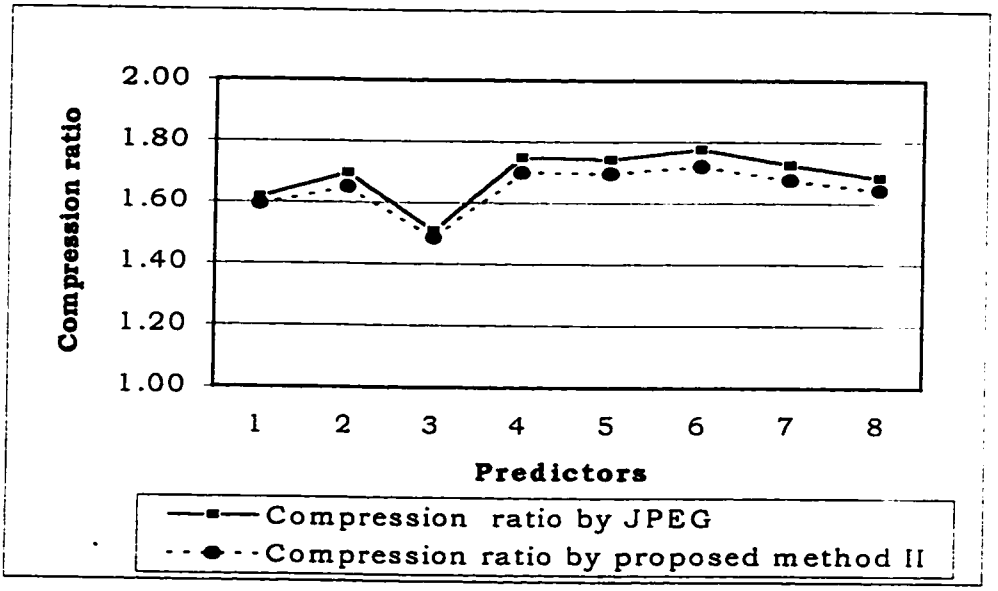


Figure D.2(f): Comparison of compression ratio for boats

References

[1] William B. Pennebaker and Joan L. Mitchell, "JPEG - Still Image Data Compression Standard", *Van Nostrand Reinhol*, New york, 1993.

[2] Gregory K. Wallace, "The JPEG Still Picture Compression Standard", *IEEE Transactions on Consumer Electronics*, vol 38, no 1, p xviii - xxxiv, February 1992.

[3] Ronald B. Arps and Thomas K. Trung, "Comparison of International Standards for Lossless Still Image Compression", *Proceedings of The IEEE*, vol 82, no 6, p 889 -899, June 1994.

[4] Stephen Wong et. al, "Radiological Image Compression -- A Review", *Proceedings of the IEEE*, vol 83, no 2, p 194-218, February 1995.

[5] Klaus Holtz, Eric Holtz , "Lossless Data Compression Techniques", *Wescon conference Record 1994*, Wescon, Los Angeles, CA, USA, RC-104, p 392-397, 1994.

[6] Koen Denecker et. al., " An Experimental Comparison of Several Lossless image Coders for Medical Image", *Proceedings of the 1997 Data Compression Conference*, p 435 ,December 1997.

[7] Koen Denecker and Peter De Neve, "A Comparative Study of Lossless Coding Techniques for Screened Continuous-Tone Images", *IEEE International Conference on Acoustic, Speech and Signal Processing – Proceedings*, vol 4 , p 2941-2944, 1997.

[8] Nasir Memon, Vishal Sippy and Xiaolin Wu, "A Comparison of Prediction Schemes Proposed for a New Lossless Image Compression Standard", *IEEE International Symposium on Circuits and Systems* , vol 2, p 309-312, 1996.

[9] James C. Tilton and Edward Seiler, "CRUSH: A Comparative Lossless Compression package", *IEEE International Geoscience and Remote Sensing Symposium* , vol 1, p 310-312, 1994.

[10] Albert P. Berg and Wasfy B. Mikhael, "A Survey of Techniques for Lossless Compression of Signals", *Midwest Symposium on Circuits and Systems* , v 2 , IEEE, p 943-946, 1995.

[11] Gopinath R. Kuduvalli and Rangaraj M. Rangayyan, "Performance analysis of reversible image compression techniques for high resolution digital teleradiology", *IEEE transaction on Medical imaging*, vol 11, no 3, pp 430-445, September 1992.

[12] P. Roos, M.A. Viergever, M.C.A Van Dijke and J.H. Peters, "Reversible Intraframe compression of medical images", *IEEE transaction on medical imaging*, vol 7, no 4, pp 328-336, December 1988.

[13] Giridhar Mandyam et al, "A two-Stage scheme for Lossless Compression of Images ", *Proceedings – IEEE international Symposium on Circuits and Systems-ISCAS 95*, Part 2 (of 3), p 1102-1105, 1995.

[14] Mccoy, J. W, Magotra N and Stearns, S; "Lossless Predictive Coding", *Midwest Symposium on Circuits and Systems, v 2, IEEE,* p 927-930, 1994.

[15] Slaven Marusic and Guang Deng , "A Study of Two New Adaptive Predictors for Lossless Image Compression", *IEEE International Conference on Image Processing* , v 2, p 286-289, 1997.

[16] Heesub Lee, "Lossless Compression of Medical Image by Prediction and Classification", *Optical Engineering* , vol 33, no 1 , p 160-166, January 1994.

[17] M, Das and J. Anand, "An efficient image compression scheme based on 2-D predictive model and a new contextual source coder", *Midwest symposium on circuits and systems*, v 2, p 913-916, 1997.

[18] Langdon, Glen G and Mealy, Bryan; "On prediction error coding methods for lossless image compression", *Asilomar conference on signals, systems & computers*, v 2, p 1442-1445, 1998.

[19] Nasir D Memon et al., "Lossless Image Compression with a Codebook of Block Scan", *IEEE Journal on Selected Areas in Communications*, vol- 13, no 1 , p 24 - 30 , January 1995.

[20] Memon, Nasir; et al., "Analysis of some common scanning techniques for lossless image coding", *Asilomar conference on signals, systems & computers*, v 2, p 1446-1450, 1998.

[21] Nasrin Tavakoli, "Lossless Compression of Medical Images", *Proceedings of the 4th annual Symposium on Computer-based Medical Systems Conference*, p 200-207, May 1991.

[22] Ho-youl Jung et al., "Rounding Transform for Lossless Image Coding ", *IEEE International Conference on Image Processing*, v 1, p 65-68, 1996.

[23] Calderbank, A.R; Daubenchies, Ingrid; et al.; "Lossless image compression using integer to integer wavelet transforms", *IEEE international conference on image processing*, v 1, p 596-599, 1997

[24] Charles G. Boncelet, "Block Arithmatic Coding for Source Compression ", *IEEE Transactions on Information Theory*, vol 39, no 5, September 1993.

[25] Marie D. Reavy and Charles G. Boncelet , " BACIC: A new Method for Lossless Bi-Level and grayscale Image Compression", *IEEE International Conference on Image Processing*, v 2, p 282-285, 1997.

[26] Wu, Nasir Memon, " CALIC - A Context based Adaptive Lossless Image Codec ", *IEEE International Conference on Acoustics, Speech and Signal Processing – Proceedings*, vol 4 , p 1890-1893, 1996.

[27] Xiaolin Wu, Nasir Memon, "Context-Based, Adaptive, Lossless Image Coding ", *IEEE Transactions on Communications*, vol 45, no 4, p 437-444, April 1997.

[28] Bruno Aiazzi, Luciano Alparone and Stefano Baronti , "A Reduced Laplacial Pyramid for Lossless and Progressive Image Communication", *IEEE Transactions on Communications*, vol 44, no 1, p 18-22, January 1996.

[29] Nasir d Memon and Khalid sayood, "Lossless Compression of RGB color Images", *Optical Engineering*, vol 34, no 6, p 1711-1717, June 1995.

[30] Nasir Memon and Khalid Sayood, "Lossless compression of video sequence", *IEEE Transactions on Communication*, vol 44, no 10, p 1340-1345, October 1996.

[31] Majid Rabbani and Paul W. Melnychuck, "Conditioning contexts for arithmatic coding of bit planes", *IEEE transaction on signal processing*, vol 40, no 1, pp 232-236, January 1992.

[32] Sei-ichiro Kamata et al., "Depth-First Coding for Multivalued Pictures Using Bit-Plane Decomposition", *IEEE Transactions on Communications*, vol 43, no 5, May 1995.

[33] George R. Robertson, Maurice F. Aburdene and Richard J. Kozick, "Differential Block Coding of Bilevel Images", *IEEE Transactions on Image Processing*, vol 5, no-9, p 1368-1370, September 1996.

[34] Mourad Abdat and Maurice g. Bellanger, " Combining Gray Coding and JBIG for Lossless Image Compression", *Proceedings of the IEEE International Conference on Image Processing*, p 851-855, 1994.

[35] Anil Kumar Chaudhary, Jacob Augustine and James Jacob, "Lossless Compression of Images Using Logic Minimization", *IEEE International Conference on Image Processing*, v 1, p 77-80, 1996.

[36] Tsong-Wuu Lin, "Image Compression using Fixed Length Quadtree Coding", *Proceeding of International Conference on Signal Processing*, p 970-973, 1996.

[37] D. A. Huffman, "A Method for the construction of Minimun Redundancy Codes", *Proc IRE*, vol 40, no 10, p 1098-1101, September 1952.

[38] Langadon G.G, "An introduction to Arithmatic Coding", *IBM Journal of Research and Development*, vol 28, no 2, p 135-149, 1984.

[39] Paul G. Howard and Jeffrey Scott Vitter, "Arithmatic Coding for Data Compression", *Proceedings of the IEEE*, vol 82, no 6, p 857-865, June 1994.

[40] Nasir D Memon and Khalid Sayood, "Asymmetric Lossless Image Compression", *Proceedings of the Data Compression Conference*, p 547, 1995.

[41] Yousef W. Nijim, Samuel D. Stearns and Wasfy B. Mikhael, "Differentiation applied to Lossless Compression of Medical Image", *IEEE Transactions on medical imaging*, vol 15, no 4, p 555-559, August 1996.

[42] Khalid Sayood and Karen Anderson, "A Differential Lossless Image Compression Scheme", *IEEE Transactions on Signal Processing*, vol-40, no 1 , p 236-241, January 1992.

[43] F, Livingston et al., "Lossless data compression in real time", *Signals, Systems and Computers; conference record of the twenty-eighth asimolar conference*, vol 2, p 1247-1250, 1994.

[44] Terry A. Welch, "A Technique for High Performance Data Compression", *IEEE Computer*, vol 17, no 6, p 8-19, June 1984.

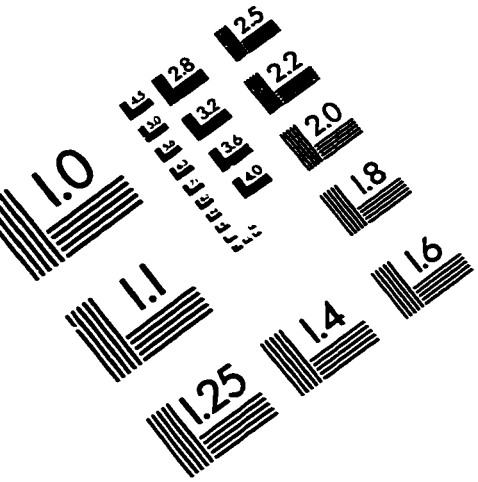
[45] Paul G. Howard and Jeffrey Scott Vitter, " Fast and Efficient Lossless Image Compression", *Proceedings of the Data Compression Conference, IEEE computer society*, p 351-360, 1993.

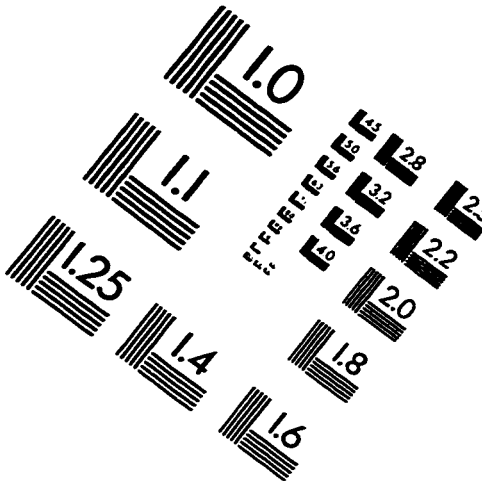
[46] Tsuhan Chen, "The Past, Present and Future of Image and Multidimensional Signal processing", *IEEE Signal Processing Magazine*, p 21-58, March 1998.

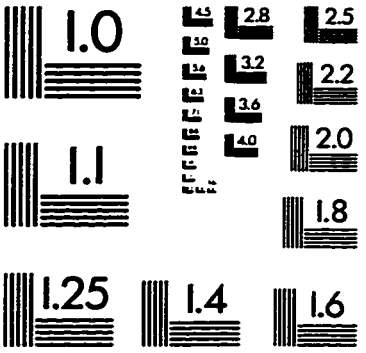
Vita

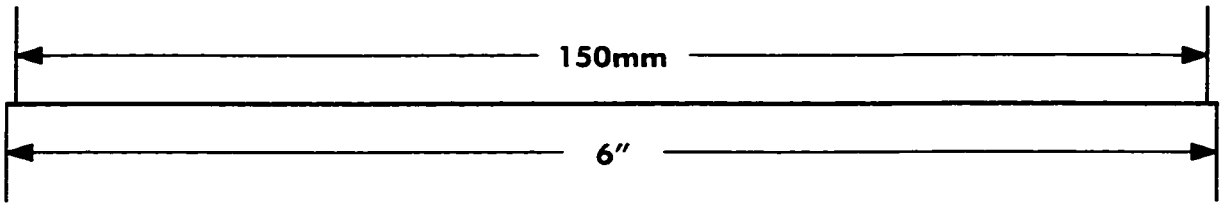
- Mohammad Sharif Chowdhury
- Born in Bangladesh.
- Received Bachelor's degree in Electrical and Electronic Engineering from Bangladesh Institute of Technology (BIT), 1991.
- Worked as an Electrical Engineer in various companies in Saudi Arabia and Bangladesh from 1991 to 1996.
- Worked as a Research Assistant in Systems Engineering department and in Electrical Engineering department at KFUPM from January 1997 to December 1998.
- Completed Master's degree in Electrical Engineering from King Fahd. University of Petroleum and Minerals, Dhahran, Saudi Arabia in December 1998.

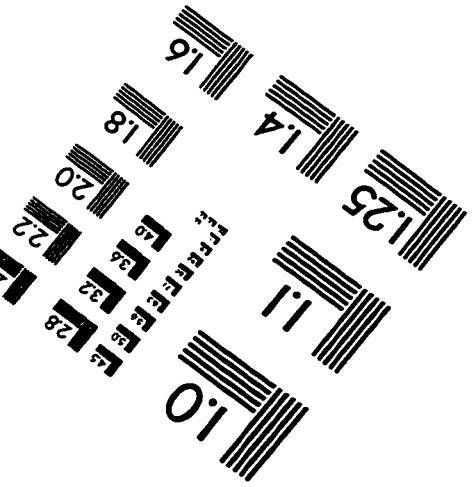
IMAGE EVALUATION
TEST TARGET (QA-3)

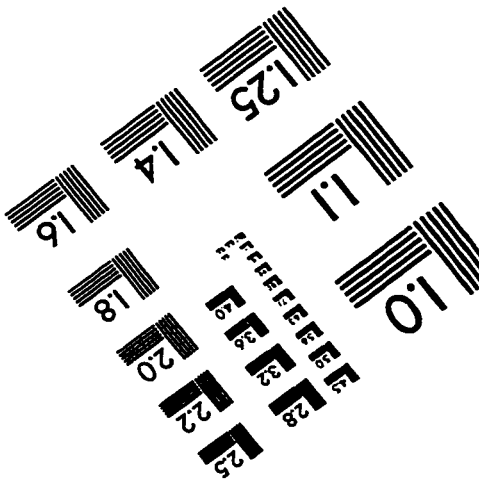












APPLIED IMAGE, Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989